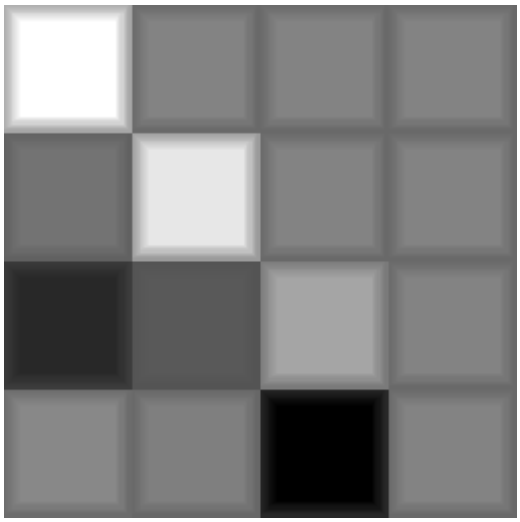
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
Санкт-Петербургский государственный университет аэрокосмического приборостроения

Балонин Н.А., Сергеев М.Б.

ОРТОГОНАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ

Учебное пособие



Санкт-Петербург
2018

Составитель: Балонин Н.А., Сергеев М.Б.

Рецензенты:

Шалыто Анатолий Абрамович
доктор технических наук, профессор
заведующий кафедрой технологий программирования СПб НИУ ИТМО

Охтилев Михаил Юрьевич
доктор технических наук, профессор
заведующий кафедрой компьютерных технологий и программной инженерии ГУАП

В пособии рассмотрены теоретические и практические методы генерации ортогональных последовательностей и матриц, опирающиеся на использование конечных полей. Использование сетевых технологий для проведения расчетов и формирования иллюстраций в форме матричных портретов позволяет студентам осваивать соответствующий математический аппарат и технологии вычислений, нарабатывать навыки создания специализированного программного обеспечении на учебном сервере livelab.spb.ru.

Пособие ориентировано на студентов, обучающихся по направлениям «Информатика и вычислительная техника», «Программная инженерия»

Работа выполнена при поддержке Минобрнауки РФ при проведении научно-исследовательской работы в рамках проектной части государственного задания в сфере научной деятельности по заданию № 2.2200.2017/4.6.

Ортогональные преобразования / учебное пособие. – СПб: ГУАП, 2018. – 57 с.

СПб: ГУАП, 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Математические сети livelab.spb.ru для студентов и mathscinet.ru для аспирантов, ориентированы на поддержку учебных занятий и научных исследований в области ортогональных преобразований информации. Это означает, что они пытаются интерпретировать запросы пользователей к решению типовых задач, записываемых как можно более просто и естественно. При этом одни формулы пользователи могут написать сами, а другие – выставить как интерпретируемые.

Сценарии. Сценарии отличаются от формул тем, что они не выполняются автоматически на момент открытия страницы. Для их старта генерируется специальная кнопка Run (начало вычислений). Иногда это удобно, тем более, что читатель, с кем студент имеет возможность делиться математическими находками, сможет в сценарии менять исходные данные, не нарушая самого письма.

Скриптовый сценарий выделяет тэг math: ($сценарий$), а вывод информации представлен командой alert(выводимое данное).

Для вывода данных сети http://livelab.spb.ru/gfom/index.php в окно служит оператор puts(данное). Этот оператор форматирует данные к удобочитаемому виду автоматически, однако, кроме него есть еще вывод putm(матрица).

A=[[1,2],[3,4]]; {{I=A/A}}; puts(I);

Результат: [[1,0],[0,1]]

Для вывода матрицы в графическом виде (в виде матричного портрета) служит оператор mesh(матрица), см. рис. 1. Этот оператор применяется для относительно небольших матриц, размером менее 100.

A=[[1,2],[3,4]]; mesh(A);

Результат:

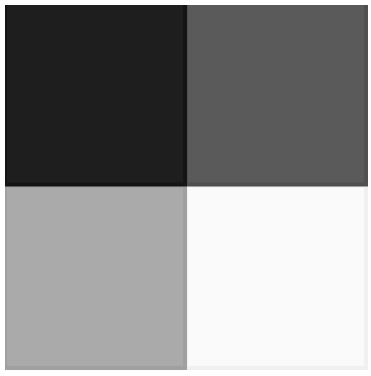


Рис. 1. Вывод портрета матрицы

Циклическая матрица (рис. 2) задается ее верхней строкой в виде:

a=[0,1,2,3,4]; A=circ(a); mesh(A);

Результат:

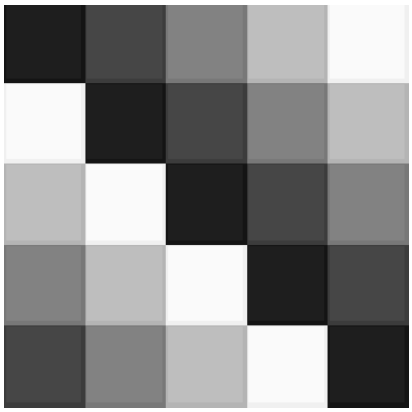


Рис. 2. Циклическая матрица

Значения элементов матрицы 1, 2, 3, 4, как видно, передаются оттенками серого, максимальный элемент (4) наиболее светлый, минимальный (1) представлен черной клеткой портрета.

Создание операций. Операции сложения и умножения в полях Галуа отличаются от обычных, поэтому их требуется конструировать в виде некоторый функций сложения и умножения. Конструирование поля Галуа GF(*p*) из такого материала, как числа, несложно, поскольку это обычные сложения и умножения целых чисел, результат – остаток по модулю *p*. Рассмотрим, например, поле GF(*p*), в программах *p* пишется как p.

```
function GF(p) { return p;
}

function gfadd(A,B,F) {
// ADD C=A+B
return (A+B)%F;
}

function gfmul(A,B,F) {
// MUL C=A*B
return (A*B)%F;
}
```

Использование библиотеки операций. Арифметические операции в полях Галуа, оформленные в виде подпрограмм, размещают в библиотеки. Рассмотрим пример с библиотечными функциями.

```
F=GF(3);
A=1; B=2; alert("A="+A+" B="+B);
C=gfadd(A,B,F); alert("C=A+B="+C);
D=gfmul(A,B,F); alert("D=A*B="+D);
```

Результат: A=1, B=2, C=A+B=0, D=A+B=2.

На учебном сервере библиотечные функции инсталлированы, их не надо самостоятельно писать. Аналогом GF(p) является более универсальный оператор gfinit(p). Приведем пример программы со сложением gfadd(A,B), вычитанием gfsub(A,B), умножением gfmul(A,B) и делением gfdiv(A,B) в поле Галуа GF(5). В программе аргументы функций заменяются на те, которые нужны по смыслу комментируемого выводом результата действия.

```
gfinit(5);
puts("'1'="+GFe);
A=3; B=4; puts("A="+A+" B="+B);
C=gfadd(A,B); puts("C=A+B="+C);
B=gfsub(C,A); puts("B=C-A="+B);
D=gfmul(A,B); puts("D=A*B="+D);
B=gfdiv(D,A); puts("B=D/A="+B);
```

Результат:

```
'1'=1
A=3  B=4
C=A+B=2
B=C-A=4
D=A*B=2
B=D/A=4
```

Интерпретация формул. При написании вами письма компьютер дает подсказку – вылавливает ошибки орфографии. То же самое касается формул, такая возможность реализована в блокноте продвинутой версии сети.

Если вы занимаетесь математическим исследованием, машина вам активно помогает, прислушиваясь к вашим запросам. Договоримся, что в вашем блокноте выражение (или выражения, отделяемые точкой с запятой), результат которого вы хотели бы знать, выделяется тэгами <f> выражение=? </f>. Встретив такую конструкцию, машина будет пытаться подставить вместо знака вопроса ответ.

Начнем с простого: <f>2+2=?; 5+7=?</f>, тэги машиной при воспроизведении записи снимаются. Это выглядит следующим образом: 2+2=4; 5+7=12.

Тригонометрия: *a*=1; sin(*a*)*sin(a)+cos(a)*cos(a)=1.

Если математическое выражение написано неправильно, то вместо знака вопроса не последует ничего, знак равенства не завершится вычисленным значением.

В сети такие формулы выполняются на javascript. На этом языке много чего можно выразить, существуют обильные средства поддержки, справочники, толкования.

Интерпретация матриц. Математическая сеть выполняет матричные операции, если они заключены в фигурные скобки. Проверим это делением невырожденной квадратной матрицы **A** на саму себя {{**I**=**A**/**A**}}. В итоге должна получиться единичная матрица **I**. Смотрите, матрицу задаем построчно:

матричная алгебра: A=[[1,2.5],[3,4]]; {{I=A/A}}; вывод I=[[1,0],[0,1]].

В программе обозначения матриц и векторов, следуя традиции, не выделяются жирным.

Наименования функций генерации стандартных матриц (генератор единичной матрицы: I=eye(3) дает I=[[1,0,0],[0,1,0],[0,0,1]], проверьте и т.п., надо ведь с чего-то начинать) Некоторые стандартные матричные функции взяты из MatLab, так проще.

Сеть старается выразить итоги расчетов как можно более точно, чтобы не злоупотреблять разрядной сеткой, числовые данные полезно форматировать (функция formats(данное)).

A=[[1,2],[3,4]]; {{I=A/A}}; alert(formats(I));

Результат: [[1,0],[0,1]]

Все эти возможности, впрочем, касаются только блокнота, в котором не надо выделять исполняемый сценарий. Так как вычисления следуют непосредственно с комментариями, эта форма заметок дает тяжеловесные страницы, аналогичные страницам с анимированными картинками. В чем-то проще и удобнее использовать обычные сценарии.

1. ПОЛЯ ГАЛУА(p^m)

Теории групп, полей, колец, как и теория чисел – это не линейно развивающаяся вдоль некоторого направления течение, или, скорее, кольцевая дорога с множеством пунктов назначения. Часть тупиков нам неинтересна. С задачей на построение ортогональных базисов, теория полей и групп превращается в конструкцию с ясным осмысленным конечным результатом (ортогональный базис).

Поле. Множество элементов, на котором определены две операции сложения и умножения (правила взяты из арифметики), называют полем.

Пояснение. *Арифметику "бесконечных" наборов чисел, изучаемую в начальной школе, позднее венчают комплексные числа, состоящие из m=2 образующих. Расширить состав образующих нелегко, впервые это обнаружил Гамильтон, пытаясь конструировать гиперкомлексные числа.*

Сносной арифметики с тремя образующими не получалось, причем за арифметику с четырьмя образующими прошлось заплатить законом коммутативности ab≠ba. Арифметика конечных наборов p чисел более примитивна по привлекаемому материалу и дает больше свободы – в ней не ограничено число образующих m (любое целое) при соблюдении ab=ba. Это арифметика векторов размера m, где m=2 – случай, восходящий к комплексным числам.

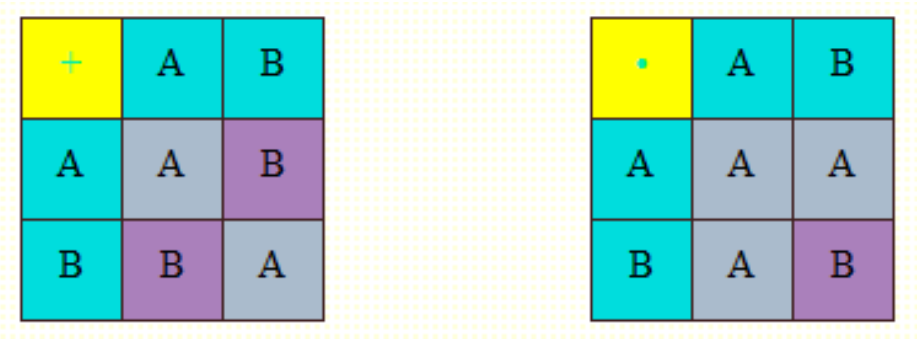
Полиномиальная интерпретация векторов как наборов коэффициентов полиномов удобна для пояснения и запоминания правил их сложения и умножения. В полиномиальной арифметике сложение и вычитание по модулю p заведомо не повышает степени полинома. Проблемно лишь умножение. Его результат, чтобы вернуть показатель старшей степени обратно, делят на некоторый неприводимый многочлен степени m и находят остаток от деления.

Поле Галуа GF(p). Поля с конечным числом элементов называют полями Галуа (Galois Fields) по имени первопроходца, Эвариста Галуа, и обозначают $GF(p)$, p – простое число.

Пример 1. Числа 0, 1, 2,…, p–1, если операции сложения и умножения выполняются по модулю p, образуют простое поле. Таблицы сложения и умножения отличаются от привычных для первого класса школы лишь тем, что содержат остатки деления на p.

Элементы поля Галуа именуются "вычетами" – имеется в виду то обстоятельство, что при выходе результата обычной операции сложения или умножения за пределы поля, он возвращается обратно последовательным вычитанием p. Не такого элемента поля, который не был бы вычетом в этом смысле. Квадратичные вычеты более редки: они образованы вычитаниями конкретно из квадратов элементов поля.

Пример 2. Поле с конечным числом элементов *GF*(2). Наименьшее число элементов, образующих поле, равно 2. Такое поле должно содержать 2 единичных элемента: A=0 относительно операции сложения и B=1 относительно операции умножения. Их можно понимать, как 0 и 1, или абстрактно, как A и B. И исследовать таблицы на предмет соблюдения, для операций с ними, всех правил арифметики, рис. 3.



+	A	B
A	A	B
B	B	A

·	A	B
A	A	A
B	A	B

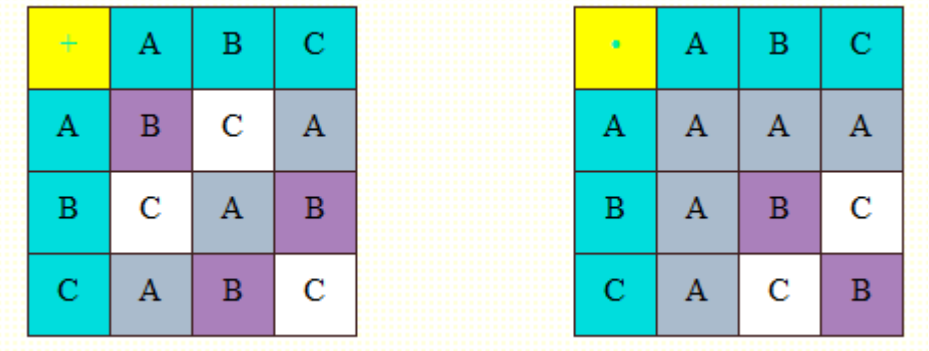
Рис. 3. Таблицы сложения и умножения в GF(2)

```
gfinit(2);
A=0; B=1; puts("A="+A+" B="+B);
C=gfadd(A,B); puts("C=A+B="+C);
D=gfmul(A,B); puts("D=A*B="+D);
```

Результат:

A=0, B=1, C=A+B=1, D=A*B=0.

Пример 3. Поле с конечным числом элементов *GF*(3) задают таблицы рис. 4.



+	A	B	C
A	B	C	A
B	C	A	B
C	A	B	C

·	A	B	C
A	A	A	A
B	A	B	C
C	A	C	B

Рис. 4. Таблицы сложения и умножения в GF(3)

Поле GF(p), p – простое число, имеет p элементов, нумеруемых как 0, 1, 2,..., p–1. Арифметика в GF(p), это арифметика по модулю p. Поэтому, обозначив элементы целыми числами, несложно сгенерировать таблицы сложения и умножения, и к ним, весь корпус арифметических функций.

```
gfinit(3);
A=1; B=2; puts("A="+A+" B="+B);
C=gfadd(A,B); puts("C=A+B="+C);
B=gfsub(C,A); puts("B=C-A="+B);
D=gfmul(A,B); puts("D=A*B="+D);
B=gfdiv(D,A); puts("B=D/A="+B);
```

Результат:

B=A*A=0, B=A*A=1, B=A*A=1

Квадратичные вычеты. Операция "умножения" не выводит целые числа поля Галуа GF(p) за пределы этого поля, по определению. Она связана с "возвращением" квадратов чисел в пределы выбранного числового диапазона, рис 5.

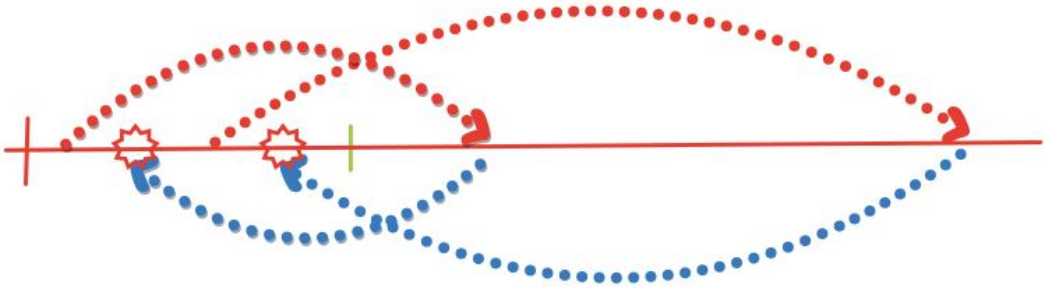


Рис. 5. Возвращение квадратов чисел в GF(p)

Те места, куда квадраты чисел "возвращаются", называются *квадратичные вычеты* (quadratic residues). Прочие числа числового поля называются *квадратичные невычеты*.

Посчитаем квадратичные вычеты поля $GF(3)$.

```
gfinit(3);
A=0; B=gfmul(A,A); puts("B=A*A="+B);
A=1; B=gfmul(A,A); puts("B=A*A="+B);
A=2; B=gfmul(A,A); puts("B=A*A="+B);
```

Результат:

B=A*A=0, B=A*A=1, B=A*A=1.

Возводить числа в степень можно, разумеется, и в более, чем во вторую, причем операции "возвращения" накроют, в таком случае, все числа. В том случае, когда для покрытия элементов "рикошетами" хватает степеней одного числа, поле, по сути дела, из них и образовано. Такое число именуют *порождающим* группу. Число элементов группы называется ее порядком.

Поле Галуа $GF(p^m)$. Арифметика сложных полей – арифметика гиперкомлексных чисел, представимых векторами с m элементами, сложение которых происходит в более простом обслуживающем операции сложного поля поле $GF(p)$. Знакомый нам пример дает поле комплексных чисел, каждое число $a+jb$ представимо парой элементов, m=2.

В конечном поле $GF(p^2)$ элемент поля A=(a, b) – вектор, параметры a, b целочисленные, операции сложения элементов идут по модулю p. Операцию умножения векторов конструируют, опираясь на правила полиномиальной арифметики (в таком случае m параметров представляют собой коэффициенты полинома).

Пример 4. Попытка построить поле GF(2^2)=GF(4) из A=0, B=1, C=2, D=3 дает таблицы, из которых видно, что для элемента 2 по операции умножения отсутствует обратный (в таблице умножения отличных от нуля элементов появился нулевой элемент A), рис. 6.

+	A	B	C	D
A	A	B	C	D
B	B	A	D	C
C	C	D	A	B
D	D	C	B	A

•	A	B	C	D
A	A	A	A	A
B	A	B	C	D
C	A	B	A	B
D	A	D	C	B

Рис. 6. Таблицы с дефектом умножения по месту A=0.

Таблицу умножения поля GF(4) можно исправить. Над полем GF(2) есть только один неприводимый многочлен X^2+X+1, соответственно, GF(4) = GF(2)[X]/X^2+X+1.

Пусть A=0, B=1, C=X, D=1+X. Эксплуатируя правила сложения и умножения полиномов расширенного поля, приводят следующие таблицы сложения и умножения (с точностью до символов, мы построили их ранее), рис. 7.

+	A	B	C	D
A	A	B	C	D
B	B	A	D	C
C	C	D	A	B
D	D	C	B	A

·	A	B	C	D
A	A	A	A	A
B	A	B	C	D
C	A	C	D	B
D	A	D	B	C

Рис. 7. Таблицы сложения и умножения в GF(4)

Первая из таблиц носит название четвертной группы Клейна (классификация от 1884 г.), обозначают V_4. Любая перестановка элементов B, C, D не меняет этой таблицы в целом – группа переходит в себя (то есть осуществляет некоторый гомоморфизм, к тому же все перестановки обратимы, поэтому – это гомоморфизм, являющийся биекцией, то есть изоморфизм, да к тому же это еще изоморфизм группы в себя, то есть автоморфизм). При автоморфизме нейтральный элемент A всегда должен переходить в себя, никаких других автоморфизмов быть не может, S_3 (симметрическая группа перестановок трех элементов) и есть искомая группа автоморфизмов.

Поле с конечным числом элементов $GF(4)=GF(2^2)$. Это арифметика векторов (a, b), компоненты принимают значения 0, 1. Поле содержит, следовательно, всего 4 элемента.

```
gfinit(2,2);
A=[0,1]; B=[1,1]; puts("A="+A+" B="+B);
C=gfadd(A,B); puts("C=A+B="+C);
D=gfmul(A,B); puts("D=A*B="+D);
```

Результат:

A=[0,1], B=[1,1], C=A+B=[1,0], D=A*B=[1,0].

Группа. В более узкой по материалу, чем теория поля, теории групп выделяют бинарную операцию, оставляя во внимании единичку, обратный элемент и ассоциативность (независимость порядка применения операции к элементам $a(bc)=(ab)c$. Таковы целые числа с операцией сложения.

Группа, в которой любые два элемента коммутируют $ab=ba$, называется *коммутативной* или абелевой. Теория полей и теория групп смыкаются – важнейшим свойством конечных полей является то, что множество всех ненулевых элементов конечного поля образует группу по операции умножения, т. е. *мультипликативную группу* порядка $p-1$.

Пример 5. Квадратичные вычеты очевидно не годны в роли элементов, порождающих группу максимального порядка. Проверим в роли порождающего $GF(3)$ группу максимального порядка элемента число 2.

```
gfinit(3); A=2;
B=gfmul(A,A); puts("B=A*A="+B);
B=gfmul(B,A); puts("B=B*A="+B);
B=gfmul(B,A); puts("B=B*A="+B);
```

Результат:

B=A*A=1, B=B*A=2, B=B*A=1.

Элементы поля, без нуля, – основа мультипликативной группы, циклической. Поскольку элементы – остатки, их называют вычеты или классами вычетов (не путать с квадратичными).

Циклические группы. Мультипликативную группу поля образует поле Галуа GF(p) без нулевого элемента. Эта группа G=GF(p)*, звездочка свидетельствует об удалении 0-й, является *циклической*, т.е. в ней есть порождающий элемент, а все остальные получаются возведением в степень порождающего. Любая циклическая группа – абелева, т.е. ее операция коммутативна.

Замечание: в высшей алгебре разница между умножением и сложением относительна, при сложении понятие "степень" n элемента a циклической группы имеет вид na. Порождающим элементом набора целых чисел будет 1.

Элементы показательной функции x^k в теории групп называют косетами (со наборами), по-нашему это элементы *прогрессии*. Векторные элементы x^{at+k}, функции от t, называют циклотомическими косетами (орбитами, в сходной интерпретации).

Примитивные элементы. Возведение в степень обнаруживает различие между элементами поля Галуа. Как можно заметить, если возводить в степень числа 3 либо 5 в поле Галуа GF(7), мы получим все элементы поля, кроме 0. Такие числа (которые порождают все элементы) называются *примитивными элементами*.

```
gfinit(7);
A=3; puts("A="+A);
A2=gfmul(A,A);  puts("A^2="+A2);
A3=gfmul(A2,A); puts("A^3="+A3);
A4=gfmul(A3,A); puts("A^4="+A4);
A5=gfmul(A4,A); puts("A^5="+A5);
A6=gfmul(A5,A); puts("A^6="+A6);
```

Результат:

A=3, A^2=2, A^3=6, A^4=4, A^5=5, A^6=1.

Исключив нулевой элемент и взяв за основу примитивный элемент, получим *циклическую* группу GF(p)*. Элементы циклической группы – степени A, A^2, ..., $A^{p-1}=1$, поэтому ее обозначают также <A> или F\{0}, F – конечное поле.

Как видно по GF(7), примитивный элемент не уникален.

Примитивному элементу ставят в соответствие *минимальный* (по порядку) полином, корнем которого он является. Такой полином с коэффициентами из поля Галуа – задает связь, которая позволяет выражать старшие степени примитивного элемента через младшие.

Теорема Ферма. Поведение прогрессий x^k описывает широко известная в теории конечных полей и мультипликативных групп теорема Ферма (родственница великой теоремы Ферма), согласно которой $x^{p-1}=1$, p – простое число. Причем при вычислении по модулю p для простых чисел всегда найдется примитивный элемент x=A, прогрессия которого займет собой все множество значений 1, ..., p–1.

Заметим, что каждый ненулевой элемент конечного поля GF(p) – корень из 1, поскольку $x^{p-1}=1$ для любого ненулевого элемента GF(p).

Допустим, x – не примитивный элемент. В таком случае прогрессия заканчивается раньше $x^n=1$, где n – делитель p–1 (т.к. сохраняется и условие $x^{p-1}=1$).

Иллюстрация прогрессий в виде строк матрицы дана на рис. 8. Для не простых n, задающих вычисление по модулю, длины прогрессий x^k уменьшаются (они меньше n–1), мы их можем видеть наяву, помещая в виде строк в матрицу. Номер строки отвечает x, номер окрашенной клетки колонки отражает x^k.

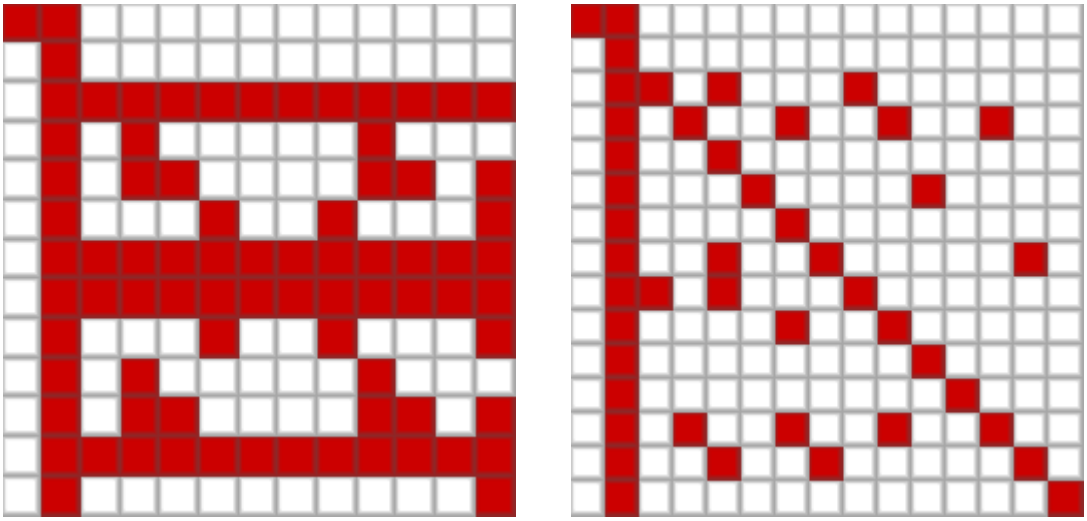


Рис. 8. Прогрессии x^k для p=13 и v=15

Хорошо видно, что для GF(13) мы имеем 4 примитивных элемента, а для составного числа v=15 ни одна прогрессия не имеет длины 15.

Собственно, нет и поля с показателем 15. Максимальная длина циклической подгруппы здесь 5. Изучите прогрессии (орбиты) для p=9 и p=27, среди них есть цепочки, регулярно прерываемые через 3. Целые числа, это не тот материал, из которого строятся GF(9) и GF(27) латанием этих дыр.

Функция Эйлера φ(n) равна количеству чисел, взаимно простых с n. Она равна p–1 для всех простых чисел n=p (мало информативна), но для не простых – уменьшается, для числа v=15 она равна 8.

Орбиты. Циклические подгруппы x^k и подкольца (образуемые добавлением 0), называемые **орбитами**, позволяют изучить состав и характер множеств элементов, входящих в мультипликативную группу кольца Z^*_v.

Орбитой элемента m кольца Z^*_v называется итог масштабирования подгруппы H (подкольцо): Orb(m)=mH. Умножение на m=0 дает частную орбиту {0} (орбиту элемента 0), умножение на m=1 дает саму подгруппу H в качестве тривиальной орбиты, на m=2 – следующую частную орбиту и т.п. Частные орбиты описывают циклические подкольца кольца.

Мы говорим о кольце, поскольку в орбиту включают 0. Рассмотрим программу, иллюстрирующую свойства орбит постепенным изменением множителя *m*.

```
v=13;
X=zero(v); puts('mod '+v);
for (m=0;m<v;m++) { // ORBITS
 x=mod(mulp(m,orbit(v,8)),v);
 puts('m='+m+' x='+8+' m*x^k='+x);
 X[i]=ds2a(v,x);
}
mesh(X);

function orbit(v,a) {
var i,x,y; x=[1]; y=a;
for (i=2;i<v;i++) {
  x=x.concat(y); y=y*a; y=y%v;
  if ((y==1)||(y==0)) break; }
return x;
}
```

Результат: в поле GF(13) вид циклической последовательности mx^k, x=8, зависит от масштабного множителя *m*, отвечающего номеру строки матричного портрета на рис. 9. В каждой строке значения, принимаемые последовательностью, отвечают номеру помеченной изменением цвета колонки.

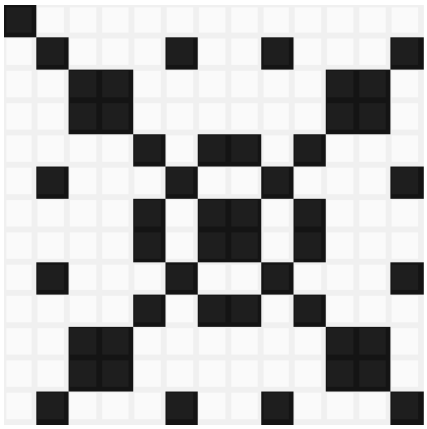


Рис. 9. Орбиты mx^k, x=8, в поле GF(13)

Масштабирование на элементы Z^*_v подгруппы H называется также *действием подгруппы* на кольцо. Орбита подгруппы Orb(H) – обозначение совокупности элементов всех частных орбит, ее и называют для большей простоты, *орбитой* H.

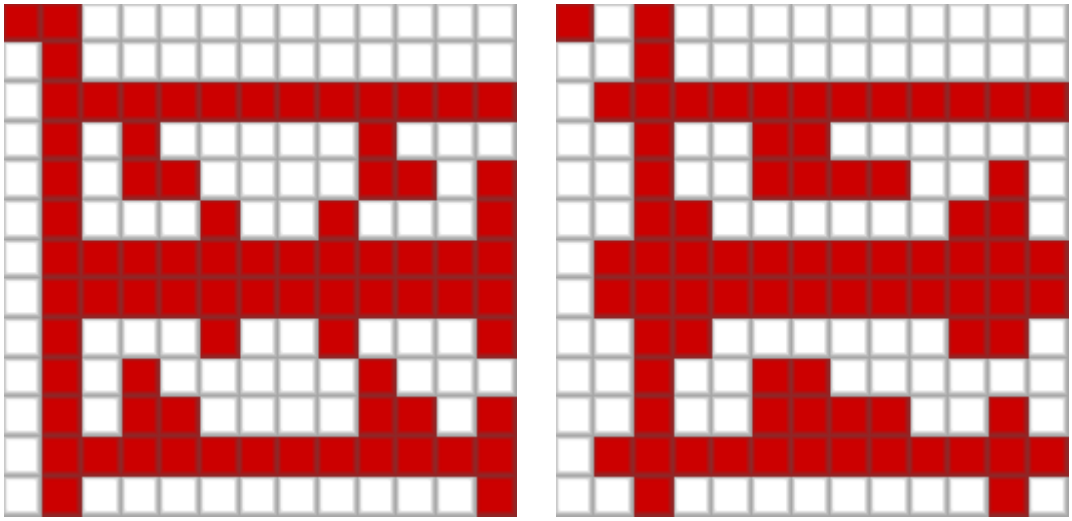


Рис. 10. Подгруппы в $Z^*_{v=13}$, т.е. орбиты m=1 и орбиты m=2

Мы видим, что умножение на разные множители меняет значения элементов mH (элементов строк рисунков).

Как обычно, интересуются *неподвижными точками* $mh=m$, h – элемент H.

Равные между собой $hm=m$ точки (инварианты действия) можно считать по-разному, в подгруппе H (замораживая m) или на множестве Z^*_v (замораживая h): получая элементы стабилизатора и фиксатора.

Стабилизатор. Равные в результате действия $hm=m$ между собой элементы H образуют *подгруппу в* H, называемую *стабилизатор* Stab(m).

Фиксатор. Равные в результате действия $hm=m$ между собой элементы Z^*_v образуют фиксатор элемента подгруппы Fix(h).

Если орбиты двух элементов Z^*_v пересекаются, то они тождественны между собой (орбиты разбивают H на классы эквивалентности).

Для H=$\{8^k\}$ в $Z^*_{v=13}$ число классов эквивалентности равно сумме числа стабилизаторов (их 16, если посчитать) по всем элементам группы, деленной на размер этой группы (их 4), согласно *лемме Бернсайда*.

Классы эквивалентности несложно выделить визуально, масштабируя H={8^k} и строя матричный портрет, рис. 11. Они видны как различные между собой строки.

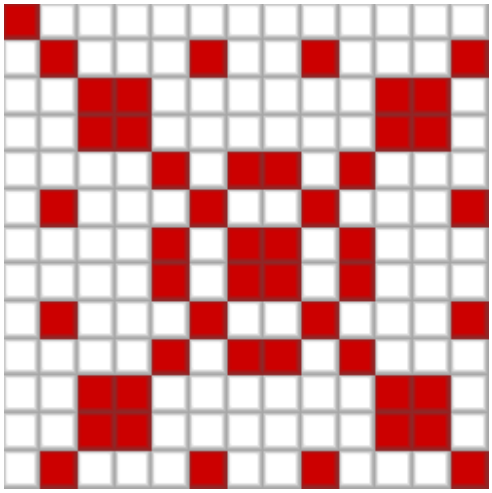


Рис. 11. Классы эквивалентности H={8^k} в $Z^*_{v=13}$

Операция умножения в полиномиальной арифметике основана на *свертке*, которая дает примерно вдвое (менее на 1, чем вдвое) больше коэффициентов, чем нужно. Вспомогательный полином, это обратная связь, с помощью которой к коэффициентам младшей части полинома произведения прибавляются старшие. Вспомогательный полином дает веса обратных связей, желательно, чтобы большую часть ветвей ампутировали нулевые коэффициенты.

Отметим. При обсуждении полиномиальной арифметики термин "простое число" заменяется термином "неприводимый многочлен" (нередуцируемый). Полином называется неприводимым, если его нельзя представить в виде двух других полиномов (конечно же, кроме 1 *и самого полинома). Полином* x^2+1 *неприводим над целыми числами, это касается также любого конечного поля. Для поля Галуа* $GF(p^m)$ *в качестве "модулей" используются трехчлены* x^m+x+1*, содержат много нулевых коэффициентов.*

Полином, который в образованном с помощью неприводимого полинома поле является генератором (степенной функции, прошивающей все поле без самопересечений), называется примитивным или базовым (элементом поля), все его коэффициенты взаимно просты.

Генератор мультипликативной группы конечного поля GF(p^m) называется *примитивным элементом* или *первообразным корнем степени* $q=p^m-1$, так как $x^q=1 \bmod p$.

Если p простое и необходимо найти все примитивные элементы поля GF(p), то достаточно найти один, а дальше возводить его в степени взаимно простые с числом $p-1$ – так получатся все примитивные элементы, их количество – значение функции Эйлера $\varphi(p-1)$.

Зависимость несложна для вспомогательных (нередуцируемых, неприводимых) полиномов, схема полиномиального умножения напоминает нейронную, рис. 12.

Примитивный многочлен над полем GF(p) именуют *минимальным многочленом* примитивного элемента поля GF(p^m). Чаще всего это $x^m=sx^d+r$, при d=1 ко всем m младшим коэффициентам свертки прибавляются все старшие с множителями s, r за исключением крайних членов, к младшему коэффициенту произведения не дотягивается ветвь с весом s, к старшему коэффициенту – ветвь с весом r.

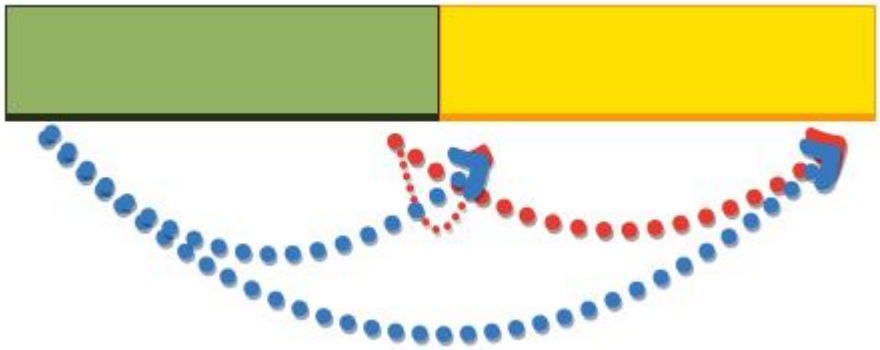


Рис. 12. Неприводимый полином дает веса обратных связей

При d>1 в этом простом алгоритме возникают поправки.

```
C=conv(A,B);
for (i=0;i<m;i++) { S=C[i];
 if (i<m-1) { S+=r*C[m+i];    if (i<d-1) S+=r*s*C[2*m-d+i]); }
 if (i>d-1)  { S+=s*C[m+i-d]; if (i<2*d-1) S+=s*s*C[2*(m-d)+i]); }
C[i]=S%p;
}
```

Потребность в s-ветви отпадает совсем для GF(p^2), p – нечетное простое, произведение пары двухкомпонентных чисел $(a,b)(c,d)=(ac–rbd),(ad+bc)$. Это поле сходно с полем комплексных чисел, где $–r$ представляет собой аналог –1 (квадрат невычета).

Параметр r=2 помимо случаев r=1, когда p+1 кратно 4, и r=3, когда p+1 кратно 6. Для более сложных полей параметры задаются таблично: таблица 1 содержит степень и весовые коэффициенты вспомогательного многочлена $x^m=sx^d+r$.

Поиск весовых коэффициентов вспомогательного многочлена использует в качестве настройки номер алгоритма GFs=s, GFr=*r*, GFd=*d*<*m* – показатель степени компоненты с параметром s.

Таблица 1. Степень и весовые коэффициенты

p	*m*	*d*	*s*	*r*	*p*	*m*	*d*	*s*	*r*	*p*	*m*	*d*	*s*	*r*
2	2	1	1	1	7	3	1	0	2	19	3	1	0	2
2	3	1	1	1	7	4	1	1	3	19	4	1	1	1
2	4	1	1	1	7	5	1	1	3	19	5	1	1	3
2	5	2	1	1	7	6	1	0	3	23	3	1	1	4
2	6	1	1	1	7	7	1	1	1	23	4	1	1	3
2	7	1	1	1	7	8	1	1	1	23	5	1	1	2
2	8	5	1	1	7	9	1	0	2	29	3	1	1	1
3	3	1	1	1	7	10	1	2	1	29	4	1	1	1
3	4	1	1	1	11	3	1	1	3	31	3	1	0	3
3	5	1	1	1	11	4	1	1	6	31	4	1	1	1
3	6	1	1	1	11	5	1	1	1	37	3	1	0	2
3	7	2	1	2	11	6	1	1	1	37	4	1	1	1
3	8	2	1	1	13	3	1	0	2	41	3	1	0	2
3	9	5	1	1	13	4	1	0	2	41	4	1	1	7
5	3	1	1	1	13	5	1	1	1	47	3	1	1	1
5	4	1	1	1	13	6	1	0	2	47	4	1	1	1
5	5	1	1	1	17	3	1	1	2	53	3	1	1	4
5	6	1	1	3	17	4	1	0	3	53	4	1	1	2
5	7	1	1	2	17	5	1	1	6	57	3	1	1	8
5	8	1	0	2	17	6	1	1	4	57	4	1	1	5

Вне таблицы [3,10,2,1,2],[3,11,2,1,2],[3,12,2,1,1][5,8,1,0,2],[5,9,4,1,4],[5,10,2,1,2],[11,7,1,1,4].

Что касается генераторов, то не занимаясь оптимизацией процесса, в качестве итерируемого можно брать элемент поля наудачу (опуская квадратичные вычеты), степенная функция искомого элемента приводит обратно не ранее исчерпания размера всего поля: $x^{p-1}=1 \mod p$, *малая теорема Ферма* (для произвольного p имеем $x^{\varphi(p)}=1 \mod p$, $\varphi(p)$ – количество чисел взаимно простых с p (меньших его), для простых чисел функция Эйлера $\varphi(p)=p-1$, что не возьми, все подходит).

Таблица и полиномиальный алгоритм умножения дают полноценную основу для построения программных систем, обеспечивающих в арифметике конечного поля решение задач на выработку матриц ортогональных преобразований. Речь об этом пойдет далее, в следующих разделах пособия.

2. ПОЛИНОМИАЛЬНАЯ АРИФМЕТИКА

Числа – материал недостаточный (при условии, что мы не будем видеть за ними такие структуры, как полиномы, речь далее идет о "полиномиальной арифметике"), для построения полей GF(p^m) с числом элементов, отличным от *простого числа*.

Простое число p называется *характеристикой* сложного поля $GF(p^m)$, образованного (например) многочленами, заданными над полем $GF(p)$ их коэффициентов, с операциями *по модулю неприводимого многочлена* $g(x)$ степени m.

Поля Галуа GF(p^m) следует понимать абстрактно, как поля гиперкомплексных чисел, задаваемых векторами из m целых чисел. Поле GF(2^2) состоит из m=2 элементов (a,b), это очень похоже на поле комплексных чисел $a+jb$. Интерпретация чисел (a,b) полиномами нам нужна для конструирования правил (таблиц сложения и умножения), с помощью которых мы будем такого сорта элементы вида (a,b) складывать и умножать.

Иными словами, полиномы сами по себе не составляют сути абстрактных полей Галуа, они нужны постольку, поскольку мы знаем, опять-таки, со школьной скамьи, операции с ними. Эти операции несколько корректируются, учитывая специфику конечных полей.

Для многих примеров нам достаточно будет поля GF(p^2) для нечетных p, оно напоминает поле комплексных чисел тем, что его элементы представимы векторами вида (a, b) с парой целых чисел (со сложением по модулю p). Этот пример рассмотрен в справочниках, и его мы реализуем ниже. Но сначала нам нужно построить простейший пример, где демонстрируется то, как работает полиномиальная арифметика.

Пример 1. Элементы поля GF(4)=GF(2^2) можно обозначить четырьмя буквами алфавита A, B, C, D (число элементов равно p^m) и дополнить их таблицами сложения и умножения с ними (таблицами Кэли). Как только вы поймете правило построения таблиц Кэли, вы сможете правильно интерпретировать и все прочие таблицы такого сорта. Для m=2 возьмем полиномы вида aX+b с парой коэффициентов, определяющих элемент поля (a,b).

При сложении полиномов степень результирующего полинома не увеличивается. Единственное отличие конечного поля, от обычного, это то, что коэффициенты a и b пары полиномов складываются по модулю p=2. Другое дело, умножение, которое порождает составляющие вида X^2. Для того, чтобы избавиться от X^2 нужно предложить любой полином второго порядка, неразложимый на произведение полиномов первого порядка. Он дает формулу, позволяющую выразить X^2 через X и 1, если X равен корню полинома A.

Над полем GF(2) есть только один неприводимый многочлен X^2+X+1, это обстоятельство записывают формально так: GF(4) = GF(2)[X]/X^2+X+1.

Полином X^2+1, например, разложим в поле GF(2), поскольку $(X+1)(X+1)=X^2+2X+1=X^2+1$. Он не подходит для миссии конструирования поля $GF(2^2)$.

Обозначим корень неприводимого полинома A (т.е. $A^2+A+1=0$), ясно, что $A^2=-A-1=A+1$ в поле GF(2). С ним мы можем построить поле $GF(2^2)$ из чисел 0, 1, A, A+1, где относительно 0, 1 работают правила поля GF(2), а для уменьшения порядка итогов операции умножения с символом A привлекается правило $A^2=A+1$. Этого достаточно для построения таблиц сложения и умножения, рис. 13.

+	0	1	A	A+1
0	0	1	A	A+1
1	1	0	A+1	A
A	A	A+1	0	1
A+1	A+1	A	1	0

·	0	1	A	A+1
0	0	0	0	0
1	0	1	A	A+1
A	0	A	A+1	1
A+1	0	A+1	1	A

Рис. 13. Таблицы сложения и умножения в поле $GF(2^2)$.

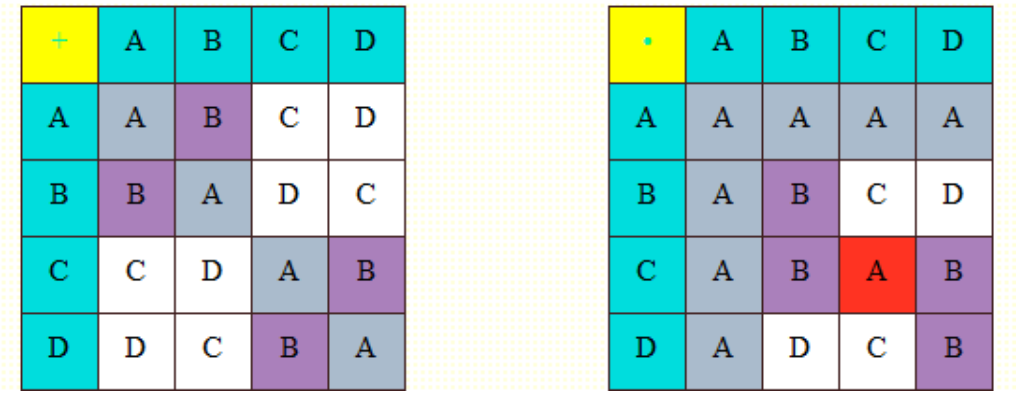
Элементы таблиц можно, в свою очередь, обозначить символами A, B, C, D, при таком обозначении на первое место выступают правила, диктуемые символьной формой записи. Таблицы Кэли можно хранить в памяти машины

```
gfinit(2,2); GFn=gfnumbers(GFp,GFm);
// TABLE OF ENTRIES
putm(GFn);
S=gfadd(GFn[2],GFn[3]);  puts("A+(A+1)="+S);
G=gfmul(GFn[2],GFn[3]);  puts("A*(A+1)="+G);
```

Результат:

```
0,0
1,0
0,1
1,1
A+(A+1)=1,0
A*(A+1)=1,0
```

Пример 2. Попытка построить поле GF(2^2)=GF(4), опираясь не на полиномы, а на числа, A=0, B=1, C=2, D=3 дает таблицы, из которых видно, что для элемента 2 по операции умножения отсутствует обратный (в таблице умножения отличных от нуля элементов появился нулевой элемент A), рис. 14.



+	A	B	C	D
A	A	B	C	D
B	B	A	D	C
C	C	D	A	B
D	D	C	B	A

·	A	B	C	D
A	A	A	A	A
B	A	B	C	D
C	A	B	A	B
D	A	D	C	B

Рис. 14. Две таблицы, слева – четвертной группы Клейна

Хотя вторая из таблиц не является таблицей Кэли (дефектна), этот пример сыграл некоторую роль в изучении групп (напомним, для построения аддитивной группы достаточно одной операции). Первая из табличек носит название четвертной группы Клейна (классификация от 1884 г.), обозначают V_4.

Любая перестановка элементов B, C, D не меняет этой таблицы в целом – группа переходит в себя (то есть осуществляет некоторый гомоморфизм, к тому же все перестановки обратимы, поэтому – это гомоморфизм, являющийся биекцией, то есть изоморфизм, да к тому же это еще изоморфизм группы в себя, то есть автоморфизм).

Таблицу умножения поля GF(4) можно исправить так, как это сделано в первом примере. Эксплуатируя правила сложения и умножения полиномов расширенного поля, построим следующие таблицы сложения и умножения (с точностью до символов, они построены ранее, GF(4) = GF(2)[X]/X^2+X+1, пусть теперь A=0, B=1, C=X, D=1+X).

Рассмотрим пример программы для расчетов в GF(4).

```
gfinit(2,2); GFn=gfnumbers(GFp,GFm);
A=GFn[0]; B=GFn[1]; C=GFn[2]; D=GFn[3];
puts("A="+A); puts("B="+B); puts("C="+C); puts("D="+D);
S=gfadd(C,D);  puts("C+D="+S);
G=gfmul(C,D);  puts("C*D="+G);
```

Результат: при элементах A=[0,0], B=[1,0], C=[0,1], D=[1,1] имеем C+D=[1,0]=B и C*D=[1,0]=B, что отвечает таблицам рис. 15.

+	A	B	C	D
A	A	B	C	D
B	B	A	D	C
C	C	D	A	B
D	D	C	B	A

·	A	B	C	D
A	A	A	A	A
B	A	B	C	D
C	A	C	D	B
D	A	D	B	C

Рис. 15. Таблицы сложения и умножения в поле GF(4)

Поля Галуа GF(p^2) для нечетных p. Этот пример наиболее интересен нам тем, что его несложно алгоритмизировать, отказавшись от хранения таблиц Кэли в памяти машины. Число элементов поля p^2 может быть большим. В таких случаях (как и в случае комплексных чисел, когда число элементов бесконечно) выгодно сконструировать не таблицы, а подпрограммы сложения, умножения (вычитания и деления).

Построение полей Галуа GF(p^2) для нечетных p облегчается гарантированным наличием неприводимых многочленов вида $X^2–r$, r – квадратичный невычет. Общее число квадратичных невычетов равно $(p–1)/2$.

Например, 2 – квадратичный невычет для p = 3, 5, 11, 13, ..., и 3 – квадратичный невычет для p = 5, 7, 17, ... Пусть $p \equiv 3 \bmod 4$, тогда p = 3, 7, 11, 19, ..., выбрав $-1 \equiv p - 1$ как квадратичный невычет, получим простой нередуцируемый многочлен $X^2 + 1$. Следовательно, элементы вида $a+jb$, где j – добавляемый элемент – элементы поля GF(p^2).

Правила сложения и вычитания наследуют правилам комплексной арифметики. Произведение $(a+jb)(c+jd)=(ac–rbd)+j(ad+bc)$.

Построим набор подпрограмм для этой арифметики.

```
function GF(p,m) {
return [p,m];
}
```

Операции умножения, сложения и вычитания реализуются элементарно:

```
function gfmul(A,B,F) {
return [(F[0]+A[0]*B[0]-A[1]*B[1])%F[0],(F[0]+A[0]*B[1]+A[1]*B[0])%F[0]];
}

function gfadd(A,B,F) {
return [(F[0]+A[0]+B[0])%F[0],(F[0]+A[1]+B[1])%F[0]];
}

function gfsub(A,B,F) {
return [(F[0]+A[0]-B[0])%F[0],(F[0]+A[1]-B[1])%F[0]];
}
```

Пример использования ее позволяет изучить операции сложных полей

```
F=GF(7,2); alert("p="+F[0]);
A=[0,1]; B=[3,2];  alert("A="+A+" B="+B); C=gfadd(A,B,F);  alert("C=A+B="+C);
```

В следующем примере конструктор поля GF(*p*,*m*) строит таблицы Кэли (Caley) сложения S и умножения M, опираясь на арифметику чисел, а для случая *m*=2 используется полиномиальная арифметика.

```
// A SIMPLE VERSION OF LIBRARY
function GF(p,m) {
var i,j,S,M;
if (m!=2) { // CAYLEY TABLES
S=matrix(p,p); M=matrix(p,p);
for (i=0;i<p;i++) for (j=0;j<p;j++) {
S[i][j]=(i+j)%p; M[i][j]=((i%p)*(j%p))%p; }
}else{ S=p; M=m; }
return [S,M];
}
```

Операции умножения, сложения и вычитания реализуются сложнее:

```
function gfmul(A,B,F) {
if (rows(F[0])) { return F[1][A][B]; }else{
return [(F[0]+A[0]*B[0]-A[1]*B[1])%F[0],(F[0]+A[0]*B[1]+A[1]*B[0])%F[0]];
}}

function gfadd(A,B,F) {
if (rows(F[0])) { return F[0][A][B]; }else{
return [(F[0]+A[0]+B[0])%F[0],(F[0]+A[1]+B[1])%F[0]];
}}

function gfsub(A,B,F) {
if (rows(F[0])) { for (var i=0;rows(F[0]);i++)
if (F[0][i][B]==A) return i; }else{
return [(F[0]+A[0]-B[0])%F[0],(F[0]+A[1]-B[1])%F[0]];
}}
```

Проктика построения программ арифметики конечных полей полезна, не пренебрегайте ею, используя только библиотечные функции.

Поля Галуа GF(*q*), *q* – степень простого числа. Постройте поля Галуа GF(27)=GF(3^3) и GF(3^6)=GF(27^2). Чтобы накрыть GF(3^6), выберем примитивный полином над полем GF(3) побольше X^6+X^5+2, что несколько усложнит правило умножения векторов в шесть компонент при работе в GF(3^6).

Неприводимый в GF(3) полином небольшой степени приведен во многих справочниках. Полезнее брать примитивный (он же неприводимый) полином X^3+2X+1, откуда имеем $X^3=X+2$. Развертку степеней в GF(3^3)* можно получить с x=[0,1,0], но если мы работаем в поле GF(p^2), p=27, такие развертки "недобирают" порядок (переходим к полю побольше).

3. ТЕОРИЯ ГРУПП И ГРУППЫ ГАЛУА

Группа. Множество элементов с некоторой бинарной операцией × образует *группу* G, если операция отвечает привычному со школьной статьи закону арифметики, касающемуся как сложения, так и умножения. Складывать или умножать между собой три числа мы можем в любой последовательности. Это свойство именуется ассоциативностью. Множество должно включать в себя нейтральный элемент (единицу, при умножении) и все обратные элементы (умножение на которые ведет к 1).

Сложность группы – понятие размытое. Допустим, мы на тарелку укладываем остальные тарелки. Если растущая горка охватывает своими комбинациями все возможные конфигурации, которые можно сложить из тарелок, это повод охарактеризовать эту ситуацию как простую, а нижнюю тарелку объявить оселком, на котором проверяется простота. Такова нормальная подгруппа, перенимающая свойство тривиального элемента порождать всю группу. В аналитической части теории групп принято рассматривать блоки (орбита элемента, фактор группа), образуемые умножениями некоторого элемента или совокупности элементов на элементы группы. Вокруг элементов или подмножеств возникает орбита или общность, рассматриваемая как конструктивный элемент группы. Что то вроде кочки с поросшей травой, основа и несомая ею часть. Сама по себе такая кочка смысла не имеет, но выделение нескольких позволяет говорить – найдена некая структура группы !

Мощность группы |G| обычно называется ее порядком (у конечного набора чисел, с операцией, это число элементов). Конечная группа содержит конечное число элементов, бинарная операция не выводит за пределы группы G×G→G. Древнейшая теорема Лагранжа отвечает представлению, возникшему еще до оформления теории групп: рассматривает размер конечной группы G как произведение размеров входящих в нее конечных подгрупп. Неполный вариант обратной теоремы к теореме Лагранжа (для некоторых делителей порядка группы G гарантируют существование подгрупп такого порядка) доказан норвежским математиком Силовом в 1872 г. Пусть $|G|=p^n s$, p – простое, s не делится на p. Тогда силовской p-подгруппой называется подгруппа G, имеющая порядок p^n. Используется для вынесения суждения о простоте группы непосредственно по ее порядку.

Централизатор элемента. В группе, где не все элементы перестановочны между собой, иногда полезно рассмотреть множество всех элементов группы, перестановочных с некоторым заданным элементом a, т.е. множество всех элементов b, для которых ab совпадает с ba ($ab=ba$). Это множество называется централизатором элемента a. У матриц выделение коммутирующих матриц **AB=BA** – путь к выделению поля (из кольца).

Легко проверить, что централизаторы сами являются группами относительно операции в исходной группе, и, значит, они образуют подгруппы данной группы. Тривиальным примером служит *централизатор нейтрального элемента* 1; он всегда совпадает со всей исходной группой, так как одно из условий, определяющих группу, в том и состоит, что каждый элемент группы перестановочен с нейтральным элементом.

Самые простые группы: циклические, в них нет никаких инградиентов, кроме 1 и всей группы, нет подгрупп. Умножая 1 на элементы циклической группы, ее же и получим. Они образованы благодаря коммутативной, т.е. абелевой, операции. Элементы могут носить имена: *a*, *aa*, *aaa* ..., и, в самом деле, при умножении и слева, и справа на *a* элемент *aa* даст *aaa*.

Группа конечна, если операция группы приводит к тому, что некоторое большое *aaa...a* = 1. Близкая по смыслу дициклическая группа состоят из двух конгломератов: *a*, *aa*, *aaa*, ...; *b*, *ab*, *aab*, *aaab*, ... с взаимными переходами (совпадениями) элементов типа $b^2=a^v$ и т.п.

Помимо правила *aaa...a* = 1 уменьшению размера конечной группы способствуют иные сходные с ним правила (позволяющие производить сокращения), но стоит иметь в виду, что с 1861 года почти столетие теория выделяла всего пять относительно сложных конструкций Матье (-io), составляемых из пар элементов *a*, *b*.

Все четные перестановки *пяти и более* букв образуют *некоммутативные простые группы*. Эти группы – их называют *знакопеременными группами* на *n* буквах – составляют второе бесконечное семейство простых групп. Кстати, именно это принципиальное различие между знакопеременными группами степени меньшей или равной 4 и степени большей или равной 5 легло в основу исследований Галуа, посвященных алгебраическим уравнениям. С его помощью объясняется принципиальная разница в свойствах решений алгебраических уравнений степени меньшей или равной 4 и степени большей или равной 5.

Экспонента и порядок группы. Заметим, что наименьшее значение показателя $m\geq 1$ элемента $a^m=1$ называют *экспонентой* группы G (наименьшее общее кратное порядков элементов конечной группы).

Инволюции (корни квадратные). Элементы порядка 2 (*aa*=1) называются *инволюциями*. По содержанию, это корни (их взаимное произведение дает 1). Легко показать, что инволюции есть в каждой группе с четным числом элементов. Поэтому, согласно теореме Томпсона–Фейта (доказательство, как всегда, сложно), всякая некоммутативная простая группа содержит инволюции.

Тривиальная группа – это группа, состоящая из одного элемента. Этот элемент обязан быть единицей группы; в зависимости от контекста его обозначают 0 (если групповая операция – сложение), 1 (если под групповой операцией подразумевается умножение) или *e*. Тривиальную группу нельзя путать с пустым множеством.

Граф Кэли. Наглядный граф переходов между элементами группы. Классификация графов по Стейнеру привела к выделению классических структур (проективные планы, аффинные планы и т.п.) и их параметров.

Интерпретации групп строго регулярными графами описываются четырьмя параметрами {ν, κ, λ, μ}. Спорадическая группа J2 представлена графом с {100, 36, 14, 12}

Абелева группа. Если от арифметики берется закон коммутативности ($ab=ba$), то группа называется коммутативной или *абелевой*.

Матрицы, например, неподходящий материал для образования абелевых групп, если иметь в виду операцию умножения. А сложение, естественно, хлопот не вызывает.

Набор целых {0, 1, ..., n–1} это группа с операцией сложения по модулю n. Обычное обозначение ее Z_n. Абелевы группы обычно записывают аддитивно (с плюсом): роль 1 играет 0, роль обратного элемента a^{-1} играет $-a$.

Основополагающая теорема о структуре конечной абелевой группы утверждает, что любая конечная абелева группа может быть разложена в прямую сумму своих циклических подгрупп, порядки которых являются степенями простых чисел. Множество всех чисел, сравнимых с a по модулю n, называется классом вычетов. На множестве целых чисел Z таких классов n, их обозначают Zn или Z/nZ. Например, Z/15Z (с модульной арифметикой) может быть разложено в прямую сумму двух циклических подгрупп порядков 3 и 5: $Z_{15}=\{0;5;10\}+\{0;3;6;9;12\}$. То же можно сказать про любую абелеву группу порядка пятнадцать, приходим к выводу, что все абелевы группы порядка 15 изоморфны.

Неабелевы группы. Если группа не абелева, то $(ab)^{-1}=b^{-1}a^{-1}$, как у матриц. Порядок здесь имеет значение (перчатки, их все равно в каком порядке снимать, а вот пиджак снять раньше пальто – проблематично). Такая арифметика, она ничуть не затруднительна и привычна. Изучение не абелевых групп связано с формализацией понятия нейтрального элемента, роль которого начинают играть нормальные подгруппы. Если их несколько, то выясняют элементы, которыми нормальные подгруппы окружены (фактор группы).

Пример *некоммутативной группы* (не абелевой) – группы всех перестановок S_n, при $n\geq 3$; множество с n элементами порождает $n!$ перестановок, как известно из школьной программы. Они названы симметрическими, поскольку впервые возникли в задачах с симметрическими многочленами типа $x_1^2+x_2^2$, симметрия которых состоит в специфичном вхождении аргументов. При $n>4$ симметрическая группа неразрешима.

Любая конечная группа изоморфна некоторой группе перестановок (теорема Кэли). Ее можно записать в виде двойной строки, где верхняя строка соответствует аргументу, а нижняя строка – значению перестановки (123/321). Перестановка называется четной, если число транспозиций (переносов) четно, такие перестановки образуют подгруппу A_n.

Подгруппа. Понятие группы связано с множеством и с операцией, над элементами множества. Если внутри множества G есть подмножество H, образующее группу с той же операцией, то выхода нет, его (с операцией) следует величать *подгруппой*. Для групп G, H не принято пользоваться обозначениями вложений (как у множеств), пишется более операционно $H \le G$. Что назвали *надгруппой* $G \ge H$, дойдите до этого сами.

<X> – обозначение наименьшей подгруппы в G, содержащей порождающее ее множество X (пересечение всех подгрупп, содержащих X).

Нормальная подгруппа. Допустим, мы умножили элемент подгруппы на масштабный множитель слева и поделили на него же справа. Если элемент после этого остался в пределах подгруппы, его "увело", но не очень, то перед нами нормальная подгруппа.

Собственно, перед нами формализованное свойство 1 мультипликативной группы. Для не абелевых групп различать элементы, похожие на единицу (нормальные подгруппы), полезно. У нормальных подгрупп меньше проблем с операцией масштабирования (смещения, в аддитивном исполнении), например, левые и правые косеты совпадают.

Фактор группа. Масштабируем элементы нормальной подгруппы H умножением их на элементы группы G. Умножение на элементы из H не выводит произведение элементов нормальной группы за пределы H.

Эта "масштабированная" нормальная группа (играющая роль 1) и называется фактор группой. Фактор группа больше нормальной подгруппы. Иными словами, если H – нормальная подгруппа группы G, тогда сэт $G/H = \{xH \mid x \in G\}$ формирует группу с операцией $(xH)(yH)=xyH$, называемую фактор группой ("умножаем множители", потом "применяем" для "масштабирования").

Коммутант группы. Подгруппа (производная группы) $G^{(0)} = [a,b]=a^{-1}b^{-1}ab$, a и b – элементы группы G.

Коммутант группы – нормальная подгруппа. Всякая подгруппа, содержащая коммутант группы – нормальная. Цепочка производных стабилизируется на *совершенной подгруппе*, коммутант которой не меняется при "взятии" производной. Если эта группа тривиальна, исходная группа G называется *разрешимой*. Для абелевых групп коммутант совпадает с единичным элементом.

Отметим. В теории колец используется иное определение коммутанта, чьи элементы – коммутаторы – определены через разности $ab–ba$.

Косеты. Когда мы масштабируем множителями показательные функции (циклические группы), построенные от чисел, мы не задумываемся, с какой стороны мы приписали множитель. Масштабирование (при мультипликативной операции группы), в аддитивном толковании, это смещение добавкой элемента.

Когда мы начинаем задумываться, мы говорим о левых или правых косетах (cosets), буквально – “со сэты”, синхронизированные масштабированием или смещением сэты.

Циклическая группа. Группа называется *циклической*, если она порождена одним элементом. Т.е. степени этого *образующего* элемента g (взаимные произведения, степени) и есть элементы группы. Последовательные повышения степени приводят к нейтральному элементу $g^n=1$, показатель степени совпадает с порядком группы $n=o(g)$.

Группа порядка p, где p – простое число, циклична (поскольку порядок элемента, отличного от единицы, не может быть равен 1, все элементы, кроме единицы, имеют порядок p, и значит, каждый из них порождает группу).

Порядок произвольного элемента циклической группы g^m равен $n/\gcd(m,n)$, где $\gcd(m,n)$ – наибольший общий делитель.

Группа G называется *периодической* (группой кручения), если все ее элементы имеют конечный порядок (образуют конечные циклические подгруппы). Всякая группа конечной экспоненты – периодическая. Произведение элементов конечной абелевой группы имеет порядок, не выше 2 (группа экспоненты 2 – абелева).

Простые группы. Группы, которые невозможно упростить менее, чем до нейтрального элемента, объединяя их элементы в субэлементы. Нельзя собрать гомоморфный образ, который вел бы себя как группа. Наименьшая некоммутативная простая группа состоит из 60 элементов. Ее можно описать как группу вращений правильного додекаэдра, переводящих его в себя. При этом каждая из 12 граней додекаэдра может занимать пять различных положений – этим и объясняется число 60.

Первый пример неабелевых простых групп был открыт Галуа. Это знакопеременная группа A_n – подгруппа индекса 2 в симметричной группе перестановок S_n, состоящая из всех четных перестановок. Простота A_n, $n\geq 5$ трактуется как неразрешимость алгебраических уравнений степени, большей 4 (теорема Галуа).

Спорадическая простая группа – это такая группа, для которой нет простого и ясного описания. Это не группа вычетов и не группа вращений, такая невнятная группа. Вместо слова "невнятная" используют термин "спорадическая". Пять примеров спорадических простых групп в 1861 году дал Эмиль Матье. В 1965 году, спустя век, Звонимир Янко (cross-section method) обнаруживает группу из $2^3\times3\times5\times7\times11\times19=175560$ элементов, обозначаемую J1.

Группа $J_2=JH$ имеет $2^7\times3^3\times5\times7$ элементов, ее интерпретацию перестановками нашел Холл (есть геометрические интерпретации Холла и Жака Тита). Компьютеры позволили построить более крупные примеры с J_3, J_4. Монструозная группа порядка, выражаемого 54-мя цифрами обнаруживает удивительные связи monster moonshine с теорией чисел, теорией модулярных форм и т.п.

Внутри всякой простой группы содержатся некие меньшие конструкции, называемые централизаторами инволюций, которые помогают понять, как устроена исходная группа. В случае групп Ри централизаторы инволюций допускают представление в виде группы квадратных матриц размера 2×2, составленных из элементов конечной числовой системы, размер которой (т.е. число ее элементов) равен некоторой нечетной степени числа 3.

Например, если 3 возводится в степень 1, то соответствующая конечная числовая система состоит из трех элементов группы вычетов по модулю 3. Для доказательства одной из ранних частичных классификационных теорем требовалось показать, что группы Ри – это единственные простые группы, обладающие следующим свойством: их централизаторы инволюций допускают представление 2×2-матрицами, составленными из элементов конечной числовой системы размера p^m, где p – простое, а m – нечетное число. Первым естественным шагом к достижению этой цели была попытка доказать следующую гипотезу: если некоторая простая группа обладает указанным свойством, то размер конечной числовой системы, из которой берутся элементы 2×2-матриц, равен нечетной степени простого числа 3.

Со временем эта гипотеза была проверена для всех случаев, кроме одного: когда $p^m = 5^1$.

Янко приступил к исследованию этого исключительного случая в полной уверенности, что простой группы нужного типа с числовой системой размера 5^1, т.е. 5, не существует. Однако, несмотря на все усилия, ему не удалось исключить такую возможность и тем самым завершить доказательство гипотезы. Наоборот, потратив немало труда, он сумел показать, что если простая группа такого вида существует, то она состоит в точности из 23×3×5×7×11×19 (т.е. 175 560) элементов. Вряд ли Янко удалось бы установить столь сильный результат, если бы одна подходящая группа не маячила на горизонте. С растущим нетерпением он двинулся дальше и показал, что если такая группа существует, то она порождается двумя 7×7-матрицами, в столбцах и строках которых стоят элементы группы вычетов по модулю 11.

Если обозначить две эти матрицы через **A** и **B**, то группа состоит из всевозможных матричных произведений вида **AA**, **BB**, **ABA**, **BBAABABBB** и т.д.

Оставалось только узнать, действительно ли эта группа содержит в точности 175 560 элементов; если бы это было не так, то рассуждения Янко приводили бы к противоречию, которое он искал с самого начала. На первый взгляд кажется удивительным, что всевозможные матричные произведения, составленные из A и B, эквивалентны всего лишь 175 560 матрицам. Ведь сюда входят и произведения матриц. Общее число матриц размера 7×7 с элементами из группы вычетов по модулю 11 очень велико и, значит, произведения этих двух порождающих матриц составляют среди них лишь ничтожную долю. Тем не менее, вычисления, проведенные к тому же полностью вручную, подтвердили существование шестой спорадической группы, которая в честь Янко (Janko) называется теперь J_1.

Гомоморфизм. Приведем еще пару мудреных слов, связанных со свойствами "умножения". Возьмем два множества, поставленные в соответствие, например, матриц и их определителей. Определитель произведения матриц равен произведению определителей. Операция "произведение" реализуется у матриц и определителей отлично друг от друга, но если установлен гомоморфизм реализаций (соответствие обоих типов произведений друг другу), мы можем судить об определителе произведения матриц, *умножая определители*, а не матрицы.

В данном случае соответствие (гомоморфизм) несет полезную нагрузку.

Ядро гомоморфизма. Ядром у матриц называется совокупность линейно-независимых между собой векторов, которые матрица аннулирует **A**x=0 (проецирует в 0), обозначается **X**=ker(**A**). Иными словами, это решение матричного уравнения **AX**=0, **X**=[x_1, x_2, ...]. Ядро гомоморфизма – сходное понятие, при гомоморфном отображении нейтральный элемент переходит в нейтральный же. Если еще есть элементы, переводимые в нейтральный, то они образуют ядро гомоморфизма. Большое ядро свидетельствует о некоторой избыточности того, что отображается на "более узкое" множество (как сужение реки).

Пример ядра гомоморфизма – нормальная подгруппа, которая по ее определению конструируется как совокупность элементов, играющих роль нейтрального элемента. Те же операции порождают основанную на нормальной подгруппе фактор-группу, более широкое образование, перед нами "естественный" гомоморфизм с ядром в виде нормальной подгруппы.

Изоморфизм. Биективный гомоморфизм (в обе стороны) называется *изоморфизмом*. Любая группа G изоморфна подгруппе симметрической группы S(G) (теорема Кэли).

Возьмем тот же пример с матрицами и их определителями. Разложив определитель на сомножители, нам не узнать, какие именно матрицы умножались (так как разные матрицы могут иметь одинаковые определители). Значит, соответствие (гомоморфизм, позволявший судить об определителе произведения матриц, умножая их определители) не работает в противоположную сторону, не установлен изоморфизм операций.

Автоморфизм. Если в соответствие поставлены элементы одного и того же множества (а не столь различные между собой множества, как матрицы и их определители), то гомоморфизм величают *автоморфизмом*.

По сути, теория групп связана с симметриями, присущими какой-либо системе. Представьте себе снежинку, вершины которой отстоят друг от друга на 60°. Если снежинку повернуть вокруг оси, проходящей через ее центр перпендикулярно к ее плоскости, на 60° или на число градусов, кратное 60, то ее вид в целом останется неизменным, даже если какая-нибудь вершина и изменила свое положение.

Показательная функция a^k представляет собой "спираль", которая разматывается внутри группы и рано или поздно возвращается к исходному элементу 1. Следом идет опять a. Вот и симметрия. Операция, которая оставляет общий вид фигуры неизменным в этом смысле, называется операцией симметрии (автоморфизмом). Сами автоморфизмы складываются в группу, исследование симметрии, это выделение группы автоморфизмов.

Группы Ли. Элементами групп могут быть параметрические зависимости (многообразия), являющиеся решениями дифференциальных уравнений. Классификация решений на основе их групповых свойств выдвинула работы Софуса Ли, изучавшего симметрии движений механических систем.

Кольцо, это расширенное толкование поля (а поле знакомо нам со школы по законам арифметики), при котором мы не требуем обратимости операции умножения. Таблица умножения ненулевых элементов кольца содержит 0.

Тело, это кольцо, тесно связанное с мультипликативной группой, которая образуется из него выбрасыванием ненужного группе 0.

Идеал, это более крупное, чем элементы или циклические подгруппы (показательные функции) объединение элементов кольца, характеризуемое идентификатором. Множество всех четных чисел образует идеал в кольце целых чисел, этот идеал порожден элементом 2 (умножением на 2). Как видно, идеал сопричастен и операции умножения (масштабирования элементов) и идентификатору 2 (на что именно масштабируем).

Определение и роль *идеала* кольца сходны с определением нормальной подгруппы в теории групп. Для подмножества четных чисел все немножечко проще, чем более общей ситуации, возникают свойства, на которые мы можем опираться. То же самое происходит с группой, в которой есть нормальная подгруппа.

Если мы забываем о существовании нейтрального и обратных элементов, то мы можем толковать о *подполугруппе*. Если нас волнует только существование обратных элементов, а все остальное, неважно, в том числе возможность выхода за пределы множества, то такой полуфабрикат называется *симметричным подмножеством*.

Поле. Множество элементов, на котором определены две операции сложения и умножения (правила взяты из арифметики), называют полем. Поля с конечным числом элементов p называют полями Галуа по имени их первого исследователя (Galois) и обозначают $GF(p)$ или, в более общем виде, $GF(p^m)$.

Мультипликативную группу поля образует поле Галуа без нулевого элемента. Такую группу обозначают как $GF(p^m)$*. Эта группа является циклической, то есть в ней есть порождающий элемент, а все остальные получаются возведением в степень порождающего.

Операция "умножения" не выводит целые числа поля Галуа GF(p) за пределы этого поля, по определению (это и есть основание, для построения группы). Она связана с "возвращением" квадратов чисел в пределы выбранного числового диапазона. Те места, куда квадраты чисел "возвращаются", называются *квадратичные вычеты* (quadratic residues). Прочие числа числового поля называются *квадратичные невычеты*.

Возводить числа в степень можно, разумеется, и в более, чем во вторую, причем операции "возвращения" накроют, в таком случае, все числа. В том случае, когда для покрытия элементов "рикошетами" хватает степеней одного числа, поле, по сути дела, из них и образовано. Такое число именуют *порождающим* (а минимальное число – порядком группы). Элементы поля, без нуля, – основа мультипликативной группы, циклической. Поскольку элементы – остатки, их называют вычеты или классами вычетов (не путать с квадратичными).

Дициклическая группа. Генерируется не одним, а парой элементов, удовлетворяющих следующим требованиям

$$\mathrm{Dic}_n = < a, x \mid a^{2n}=1, x^2 = a^n, x\, a\, x^{-1} = a^{-1} >$$

Группа Dic_n имеет порядок $4n$ и содержит пару циклических подгрупп порядков $2n$. Каждый элемент ее однозначно представлен как $a^k x^l$, где $k < 2n$ и $l = 0,1$.Половина элементов группы, это a^k (или $[k,0]$), другая половина $a^k x$ (или $[k,1]$).

Число таких элементов: $4n$ – порядок группы. От комплексной арифметики указанных кватернионов, можно перейти к арифметике степеней, причем работают следующие правила упрощения $a^k a^m = a^{k+m}$, $a^k a^m x = a^{k+m} x$, $a^k x a^m = a^{k-m}$, $a^k x a^m x = a^{k-m+n} x$.

Как только появились у нас обозначения, для элементов, с правилами арифметики, т.е. с операцией умножения, так у нас появилась возможность производить операции в группе. Какие операции? У группы выбор небольшой, создадим библиотеку .

```
function Dicmul(A,B,n) { var a,b;
if (A[1]==0) { a=A[0]+B[0]; b=B[1]; }else{ a=A[0]-B[0];
if (B[1]==1) { a=a+n; b=0; }else{ b=1;}}
return [(2*n+a)%(2*n),b];
}
function Dicpow(A,m,n) { var B=equal(A);
for (var i=1;i<m;i++) B=Dicmul(B,A,n);
return B;
}
```

Воспользуемся ею

```
n=2; n2=2*n; n4=4*n; puts("2n="+n2); a=[1%(2*n),0]; x=[0%(2*n),1];

x2=Dicmul(x,x,n); puts("x^2="+x2);
an=Dicpow(a,n,n); puts("a^n="+an);
```

Результат: 2n=4, x^2=[2,0], a^n=[2,0].

Вычисляя цепочки степеней элемента, получаем информацию о мультипликативной структуре группы, видны циклические подгруппы порядков 2*n* и 2.

```
n=2; n2=2*n; n4=4*n; puts("2n="+n2); a=[1%(2*n),1]; S=true;
for (m=1;m<10;m++) if (S) {
b=Dicpow(a,m,n); if (m>1) S=!Diceq(a,b);
if (S) puts("a^"+m+"="+b);
}

function Dicmul(A,B,n) { var a,b;
if (A[1]==0) { a=A[0]+B[0]; b=B[1]; }else{ a=A[0]-B[0];
if (B[1]==1) { a=a+n; b=0; }else{ b=1;}}
return [(2*n+a)%(2*n),b];
}

function Dicpow(A,m,n) { var B=equal(A);
for (var i=1;i<m;i++) B=Dicmul(B,A,n);
return B;
}

function Diceq(A,B) {
return ((A[0]==B[0])&&(A[1]==B[1]));
}
```

Результат:

2n=4, a^1=[1,1], a^2=[2,0], a^3=[3,1], a^4=[0,0].

Пусть G = $\langle a \rangle$ – подгруппа Dic_n, генерируемая элементом a (минимальная группа, включающая элемент). Тогда G – циклическая группа.

Группа порядка $2n$ имеет индекс $[Dic_n:G] = 2$ (индекс показывает соотношение элементов). Как любая подгруппа с индексом 2, G автоматически нормальная подгруппа. Группа Dic_n/G – циклическая группа порядка 2.

Линейная комбинация элементов группы (групповое кольцо) может быть записано в форме $R=R1+R2$, $R1, R2 \subseteq \langle a \rangle$.

Отметим, что сэт D группы G называется относительным дифференциальным набором (a relative difference set, RDS) с запрещенной подгруппой N, если соблюдены условия для образования RDS.

Запрещенная подгруппа здесь $\langle x^2 \rangle$, с нею R_1, R_2 образуют составные части для нахождения двух бинарных последовательностей, с помощью которых строится матрица Адамара.

4. ОРТОГОНАЛЬНЫЕ МАТРИЦЫ

Рассмотрим поэтапно, как в конечном поле GF(p), p=4t–1, показательная функция порождает циклическую подгруппу G степеней элемента g. Подгруппа размера (p–1)/2 связана с номерами отрицательных элементов –1 последовательности a, составляющей верхнюю строку циклической матрицы **A** порядка 4t.

Нормализованная матрица Адамара **H** состоит из блока –**A** с каймой из 1, рис. 16.

```
p=7; v=(p-1)/2; gfinit(p);
g=2; x=zero(1); x[0]=g; puts("g="+g);
for (k=0;k<p-1;k++) x[k+1]=gfmul(x[k],g);
puts("Показательная функция="+x);
G=rowcol(x,0,v-1); puts("Подгруппа="+G);
a=ds2a(p,G); puts("Последовательность="+a);
A=circ(a); H=border(minp(A),1); mesh(H);
{{I=H'*H}} putm(I);
```

Результат: обратите внимание на соответствие индексов 1, 2, 4 (элементов подгруппы) номерам выделенных цветом элементов (белых у нормальной формы)

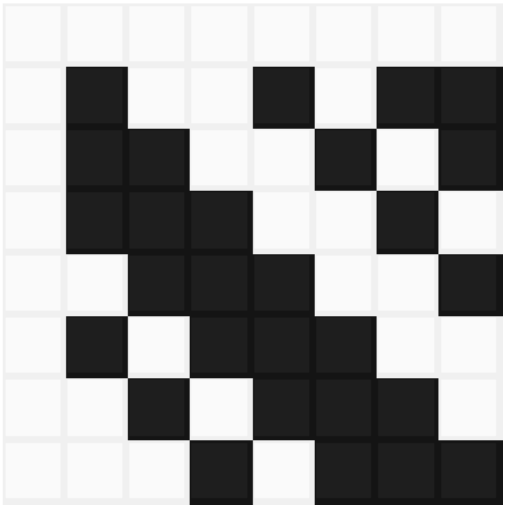


Рис. 16. Нормализованная матрица Адамара **H**

```
g=2
Показательная функция=2,4,1,2,4,1,2
Подгруппа=1,2,4
Последовательность=1,-1,-1,1,-1,1,1
```

Для нечетных простых чисел вида $4t–3$ расчет несколько меняется и завершается он построением ортогональной конференц-матрицы **C** (с 0 на диагонали), поскольку на порядках $4t–2$ матриц Адамара не бывает.

```
p=5; v=(p-1)/2; gfinit(p);
g=4; x=zero(1); x[0]=g; puts("g="+g);
for (k=0;k<p-1;k++) x[k+1]=gfmul(x[k],g);
puts("Показательная функция="+x);
G=rowcol(x,0,v-1); puts("Подгруппа="+G);
a=ds2a(p,G); a[0]=0; puts("Последовательность="+a);
A=circ(a); C=border(minp(A),0); mesh(C);
{{I=C'*C}} putm(I);
```

Результат:

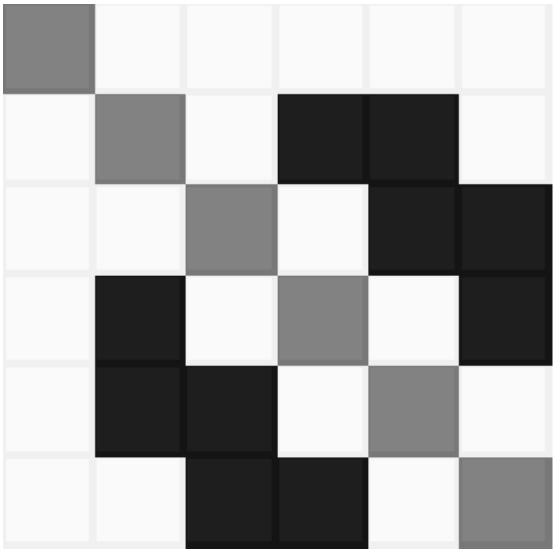


Рис. 17. Нормализованная конференц-матрица **C**

```
g=4
Показательная функция=4,1,4,1,4
Подгруппа=4,1
Последовательность=0,-1,1,1,-1
```

Отметим, что элемент, порождающий всю группу размера p, нам не годится, нам нужны номера только отрицательных элементов матрицы, а их почти вдвое меньше размера группы. Той же цели служат символы Лежандра.

Символы Лежандра, помимо стартового 0, равны 1 для квадратичных вычетов, и –1 – для невычетов. Они дают верхнюю вектор-строку циклической матрицы Якобсталя **A**, при добавлении к ней каймы образуется ортогональная по строкам и столбцам конференц-матрица **C**, порождающая, в свою очередь, матрицу Адамара **H**.

Рассчитаем квадраты элементов поля GF(p), p=4t–3.

```
p=5; gfinit(p);

a=[0,1,2,3,4]; a2=zero(a);
for (i=0;i<p;i++) a2[i]=gfmul(a[i],a[i]);
puts(a+" элементы поля");
puts(a2+" квадратичные вычеты");
```

Результат:

```
0,1,2,3,4 элементы поля
0,1,4,4,1 квадратичные вычеты
```

Расчет символов Лежандра (индексов принадлежности элементов поля к квадратичным вычетам) и циклической матрицы, основанной на них. Обратите внимание на симметрию (или антисимметрию для p=4t–1) символов относительно середины.

```
p=5; gfinit(p);

a=line(p); a2=zero(a); s=minp(one(a));
for (i=0;i<p;i++) a2[i]=gfmul(a[i],a[i]);
for (i=0;i<p;i++) for (j=0;j<p;j++) if (a[i]==a2[j]) s[i]=1;
s[0]=0; S=circ(s); mesh(S);
puts(a+" элементы поля "+a2+" квадратичные вычеты");
puts(s+" символы Лежандра");
puts(S);
```

Результат: Циклическая матрица Якобсталя со строками
[[0,1,-1,-1,1],[1,0,1,-1,-1],[-1,1,0,1,-1],[-1,-1,1,0,1],[1,-1,-1,1,0]]

0,1,2,3,4 элементы поля 0,1,4,4,1 квадратичные вычеты
0,1,-1,-1,1 символы Лежандра

Свойства циклической матрицы (матрицы Якобсталя), она почти ортогональна.

```
p=5; gfinit(p);

a=line(p); a2=zero(a); s=minp(one(a));
for (i=0;i<p;i++) a2[i]=gfmul(a[i],a[i]);
for (i=0;i<p;i++) for (j=0;j<p;j++) if (a[i]==a2[j]) s[i]=1;
s[0]=0; S=circ(s); mesh(S); puts(S);
puts("ПРОВЕРКА ОРТОГОНАЛЬНОСТИ");
{{I=S'*S}} putm(I);
```

Результат: Циклическая матрица Якобсталя с строками
[[0,1,-1,-1,1],[1,0,1,-1,-1],[-1,1,0,1,-1],[-1,-1,1,0,1],[1,-1,-1,1,0]]

Ортогонализация добавлением бордюра из единиц для порядков 5, 13, 17, ... дает конференц матрицу **C**, которая симметрична

```
p=5; gfinit(p);

a=line(p); a2=zero(a); s=minp(one(a));
for (i=0;i<p;i++) a2[i]=gfmul(a[i],a[i]);
for (i=0;i<p;i++) for (j=0;j<p;j++) if (a[i]==a2[j]) s[i]=1;
s[0]=0; S=circ(s); C=border(S); mesh(C); puts(C);
puts("ПРОВЕРКА ОРТОГОНАЛЬНОСТИ");
{{I=C'*C}} putm(I);
```

Результат: Нормализованная конференц-матрица **C**, рис. 17.
[[0,1,1,1,1,1],[1,0,1,-1,-1,1],[1,1,0,1,-1,-1],[1,-1,1,0,1,-1],[1,-1,-1,1,0,1],[1,1,-1,-1,1,0]]

Для порядков 3, 7, 11, ... и матрица Якобсталя и конференц матрица **C** кососимметричны (это отражается на знаках первого столбца и строки), рис. 18.

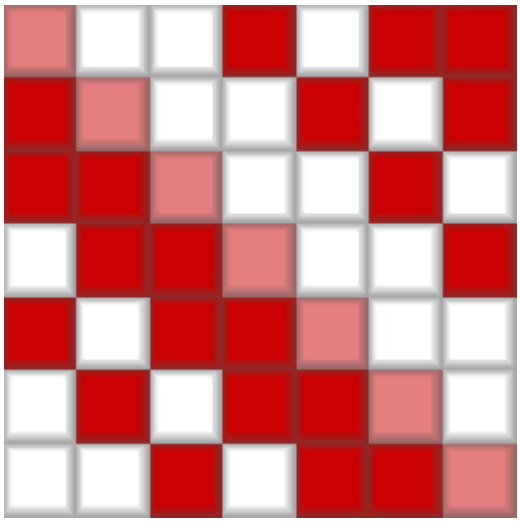


Рис. 18. Циклическая матрица Якобсталя S_7

```
p=7; gfinit(p);

a=line(p); a2=zero(a); s=minp(one(a));
for (i=0;i<p;i++) a2[i]=gfmul(a[i],a[i]);
for (i=0;i<p;i++) for (j=0;j<p;j++) if (a[i]==a2[j]) s[i]=1;
s[0]=0; S=circ(s); C=border(S,0,-1,1); mesh(C); puts(C);
puts("ПРОВЕРКА ОРТОГОНАЛЬНОСТИ");
{{I=C'*C}} putm(I);
```

Результат: Нормализованная конференц-матрица **C** со строками

[[0,1,1,1,1,1,1,1],[-1,0,1,1,-1,1,-1,-1],[-1,-1,0,1,1,-1,1,-1],[-1,-1,-1,0,1,1,-1,1],
[-1,1,-1,-1,0,1,1,-1],[-1,-1,1,-1,-1,0,1,1],[-1,1,-1,1,-1,-1,0,1],[-1,1,1,-1,1,-1,-1,0]]

Образование матрицы Адамара H из конференц матрицы C первого типа, порядок матрицы удваивается

```
p=5; gfinit(p);

a=line(p); a2=zero(a); s=minp(one(a));
for (i=0;i<p;i++) a2[i]=gfmul(a[i],a[i]);
for (i=0;i<p;i++) for (j=0;j<p;j++) if (a[i]==a2[j]) s[i]=1;
s[0]=0; S=circ(s); C=border(S,0);

I=eye(C); {{H11=C+I; H12=C-I; H22=-C-I}}
H=square(H11,H12,H12,H22); mesh(H);

puts("ПРОВЕРКА ОРТОГОНАЛЬНОСТИ");
{{I=H'*H}} putm(I);
```

Результат: Нормализованная матрица Адамара **H**, рис. 19.

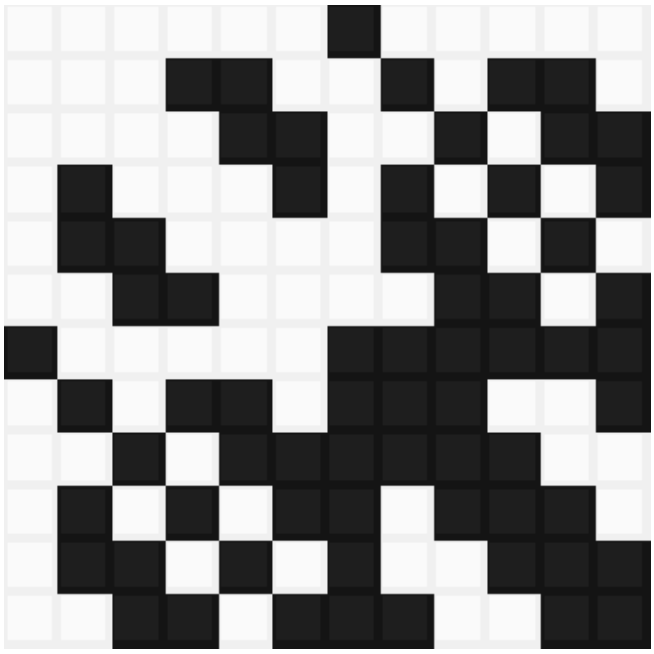


Рис. 19. Нормализованная матрица Адамара **H**

Образование матрицы Адамара H из конференц матрицы C второго типа

```
p=7; gfinit(p);

a=line(p); a2=zero(a); s=minp(one(a));
for (i=0;i<p;i++) a2[i]=gfmul(a[i],a[i]);
for (i=0;i<p;i++) for (j=0;j<p;j++) if (a[i]==a2[j]) s[i]=1;
s[0]=0; S=circ(s); C=border(S,0,-1,1);

I=eye(C); {{H=C+I}} mesh(H);

puts("ПРОВЕРКА ОРТОГОНАЛЬНОСТИ");
{{I=H'*H}} putm(I);
```

Результат: Нормализованная матрица Адамара **H**, рис. 20.

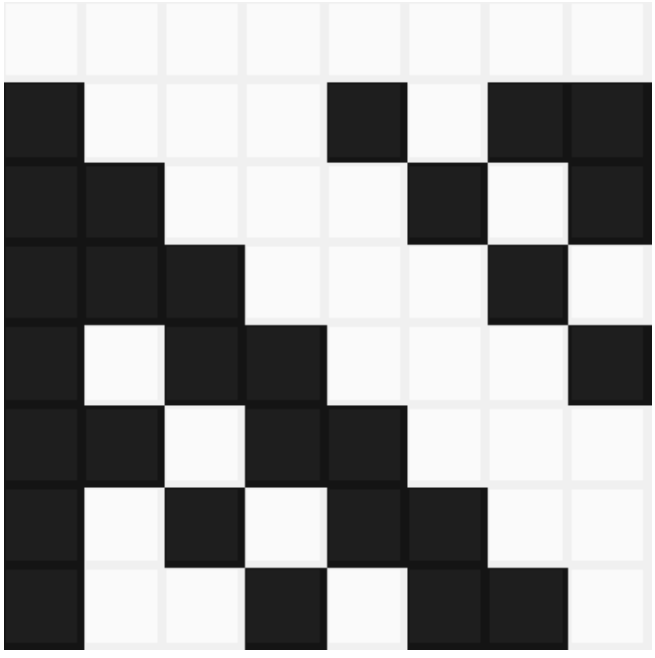


Рис. 20. Нормализованная матрица Адамара **H**

Построение блочных матриц Якобсталя и Белевича для GF(9), GF(25) связано с конструированием большого числа блоков, рис. 21.

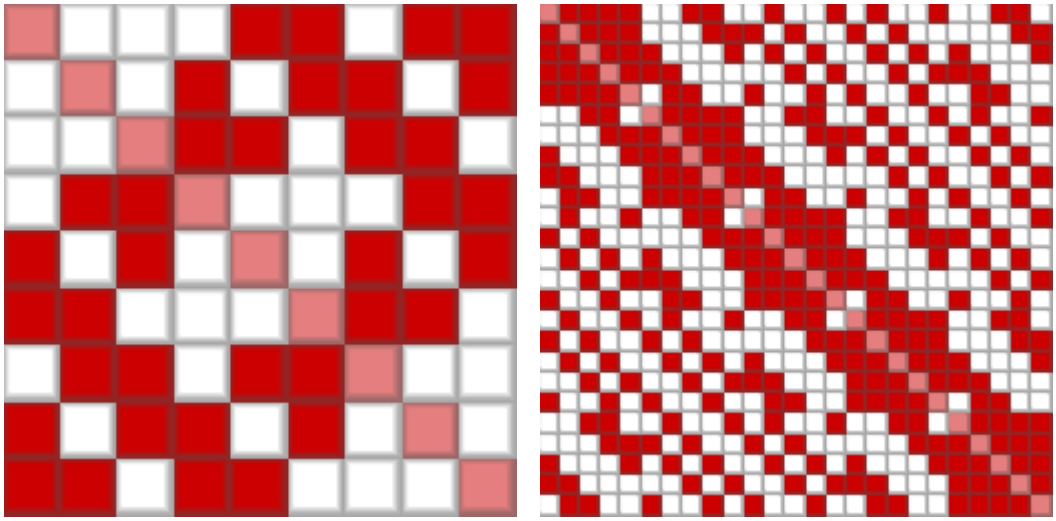


Рис. 21. Матрицы Якобсталя, построенные на блоках Лежандра для GF(9), GF(25)

Квадратичные вычеты полей Галуа GF(p^2) позволяют идентифицировать элементы полей значениями символов Лежандра на принадлежащие квадратичным вычетам 1 и невычетам –1. Цепочки символов разбиваются на фрагменты длиной p, позволяющие строить циклические блоки блочных циклических матриц Якобсталя.

```
function Jacobsthal(S,p) {
// S=[0,1,1,1,-1,-1,1,-1,-1];
if (p==3) { a=[0,1,1]; b=[1,-1,-1]; c=[1,-1,-1];
a=circ(a); b=circ(b); c=circ(c); Q=circul(a,b,c);
}
// S=[0,-1,-1,-1,-1,1,1,-1,-1,1,1,-1,1,1,-1,1,1,-1,-1,1,1,-1,1,1,1,-1,-1,1];
if (p==5) { a=[0,-1,-1,-1,-1]; b=[1,1,-1,-1,1];
c=[1,-1,1,1,-1]; d=[1,-1,1,1,-1]; e=[1,1,-1,-1,1];
a=circ(a); b=circ(b); c=circ(c); d=circ(d); e=circ(e);
Q=circul(a,b,c,d,e);
}
return Q;
}
```

Блочная матрица Якобсталя, рис. 21, использует символы Лежандра [0,1,1,1,-1,-1,1,-1,-1].

5. ТЕПЛИЦЕВЫ МОНОБЛОКИ

Для начала, вспомним, как мы вычисляем матрицу Якобсталя, порядок 7. Берется цепочка чисел W={0,1,2,3,4,5,6} и вычисляется вторая цепочка U квадратов от них по модулю 7, рис. 22. Если число первой цепочки есть во второй цепочке, ставим 1, нет, ставим –1 за исключением первого нейтрального элемента. Ставим на первое место 0. Перед нами вырисовывается строка 0,1,1,–1,1,–1,–1 циклической матрицы Якобсталя (почти ортогональной).

```
S=circ(legendre(7)); {{I=S'*S;}}; putm(I); mesh(S);
```

Результат: Циклическая матрица Якобсталя S, рис. 17. Матрица S'*S

```
6,-1,-1,-1,-1,-1,-1
-1,6,-1,-1,-1,-1,-1
-1,-1,6,-1,-1,-1,-1
-1,-1,-1,6,-1,-1,-1
-1,-1,-1,-1,6,-1,-1
-1,-1,-1,-1,-1,6,-1
-1,-1,-1,-1,-1,-1,6
```

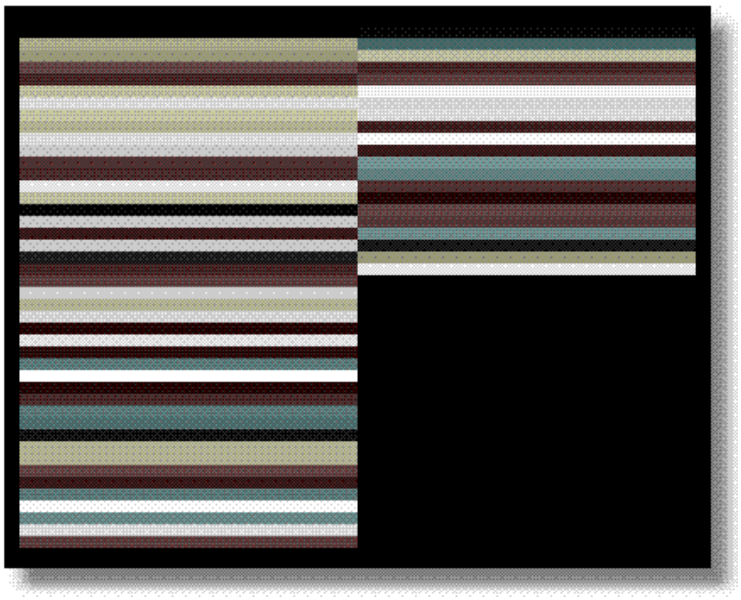


Рис. 22. Элементы цепочек W и U отображены цветными полосками

Теплицевы (негациклические) матрицы четных порядков, моноблоки, вычисляем неолько иначе. В арифметике полей Галуа, возьмем для конкретности GF(p^2), $p=n-1$, n – длина синтезируемой последовательности, та же идея выглядит следующим образом. Берутся не число и его квадрат, а две соседние степени примитивного элемента x вида $w=x^{L-1}$ и $u=x^{L}$, т.е. заведомые "нечет" и "чет". Выбрав две точки опоры, путешествие ускоряем. Оба числа возводятся в последовательные степени, образуя экспоненты $W=\{w^0, w^1, w^2, ..., , w^{n-1}\}$ и $U=\{u^0, u^1, u^2, ..., u^{n/2-2}\}$, состоящие из элементов циклических групп, второй цепочка примерно вдвое короче. В примере, с которого мы начинали, часть квадратов тоже совпадают между собой – конечное поле, оно тесное.

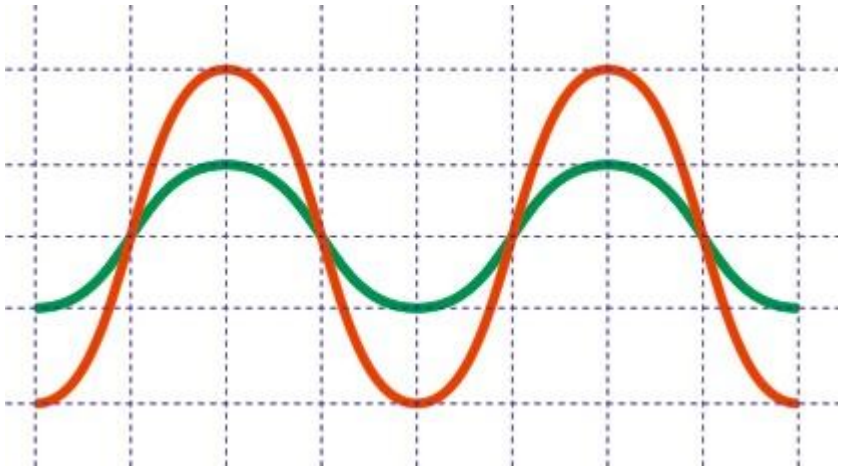


Рис. 23. Две осциллирующие зависимости, аналоги W и U

Для поля GF(p^2) характерна арифметика, близкая к арифметике комплексных чисел. Степенные функции W и U ведут себя как "синусоиды" (напомним, что синус еще Эйлер связал с разностью пары экспонент). Двумерные элементы-точки обоих можно весьма условно отобразить осциллирующими кривыми на плоскости (реальная картинка осцилляций при малом множестве точек более хаотична, конечно, показан принцип). После предустановки "начальных условий", мы запускаем процесс из двух соседних точек, далее экспоненты "осциллируют", образуя наложения. Чем выше порядок задачи, тем ближе дискретная картинка к непрерывной.

Степень $L=2n$ равна удвоенному размеру первой цепочки для приведения цепочек в "резонанс": нам важно, чтобы они не ходили рядом друг с другом вхолостую, а пересекались, , рис. 23. Показатель степени для "нечета" можно сократить вдвое или вчетверо $w=x^{n-1}$ или $w=x^{v-1}$, $v=n/2$, на разрешимости задачи это не отразится. То же самое можно делать с "четом" (но, увеличивать $u=x^{4n}$ или $u=x^{8n}$).

Перед сравнением элементов цепочек первую еще "проецируют", берут не W, а степенную функцию (trace map) от нее $W^*=\alpha W+(\alpha W)^p$, $\alpha=x^v$, $v=n/2$.

Коррекция, выводимая из условий ортогональности.

Далее сравниваем W* и U. Если в первой цепочке есть число второй цепочки, ставим 1, нет – ставим –1 (за исключением первого элемента). Перед нами строка элементов 0,1,–1,1,–1, ..., теперь уже, ортогональной негациклической матрицы Адамара, рис. 24.

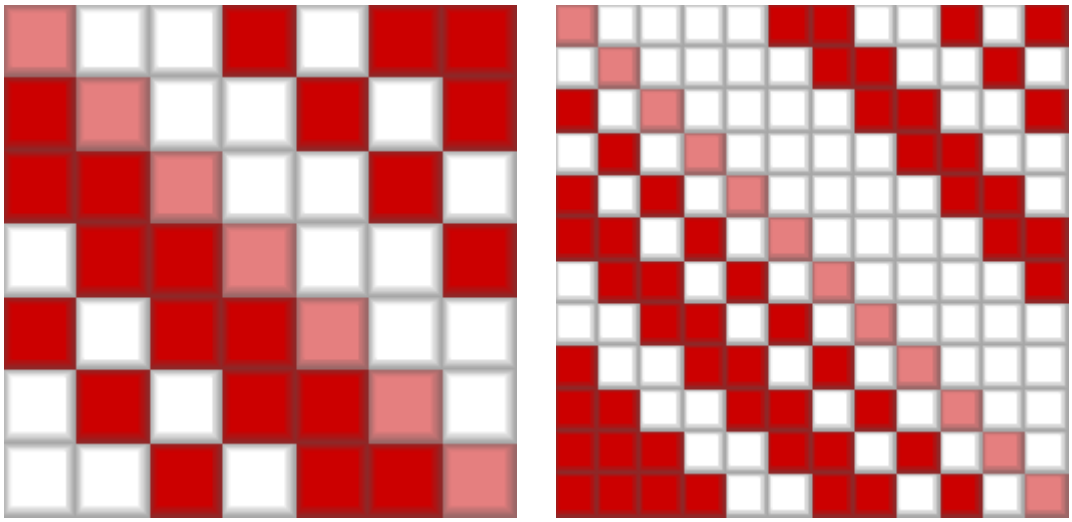


Рис. 24. Циклическая матрица Якобсталя $\mathbf{S}_7$ и негациклическая матрица Адамара H_{12}

Алгоритм ниже рестартами ищет примитивный элемент x, ориентируясь на невязку m (максимальный по абсолютному значению элемент вне диагонали $\mathbf{H}^T\mathbf{H}$, $\mathbf{H}$ – негациклическая матрица Адамара).

Отметим, что при проецировании можно менять значение α, если нам не нужна ортогональная матрица, например, когда это блок более крупной матрицы, которая ортогональна, но в целом.

Циклические матрицы Якобсталя – моноциклы, нечетного порядка. Сами по себе они не ортогональны и нуждаются в кайме. Негациклические матрицы Адамара – ортогональные моноциклы четного порядка.

Напишем программу вычисления.

```
// INTERNET MATLAB PROGRAM

if (tick==0) { n=8; v=n/2;
p=n-1; gfinit(p,2); if (p==27) gfinit(3,6);
n1=n-1; v1=v-1; M=10000; R=true;
}


for (k=0;k<10;k++) if (R) {
 x=gfrand(); L=2*n;
 U=gfexp(gfpow(x,L),v-1);
 W=gfexp(gfpow(x,L-1),n);
 W=gfmul(gfpow(x,v),W);
 W=gfadd(W,gfpow(W,p));
 a=gfeq(W,U); a[0]=0;
 H=negacirc(a); m=maxabslsm(H);
if (M>=m)  { M=m; if (M==0) R=false; }
}

// INFORMATION
puts("n="+n+" p="+p+" r="+GFr+" Lim="+M+"; tick="+tick);

// RESTART
if (M>0) { restart(0) }else{
// FINISH
puts("a=["+a+"];"); mesh(H);
{{I=H'*H}} putm(I);
}
```

Результат: . Циклическая матрица Якобсталя $\mathbf{S}_7$, рис. 24.

```
n=8 p=7 r=1 Lim=0; tick=0
a=[0,1,1,-1,1,1,1,-1];
```

6. КОЛЬЦА И ИДЕАЛЫ

Кольца, это усеченные (по количеству правил) поля (а поле знакомо нам со школы по законам арифметики) в которых мы не требуем обратимости операции умножения. Определение и роль идеала кольца сходны с определением нормальной подгруппы в теории групп. Множество всех четных чисел образует идеал в кольце целых чисел, этот идеал порожден элементом 2.

Сложение и умножение в поле всегда можно пустить вспять, перейдя к вычитанию и делению. В отличие от теории групп (где берется одна операция, по определению), для элементов кольца определены дополнительные операции. Простейший пример, это кольцо целых чисел. Любые два целых числа можно сложить или вычесть друг из друга без ограничения. То же самое касается их умножения, но вот разделить целые числа не всегда возможно.

Пример 1. Попытка построить поле $GF(2^2)=GF(4)$ из A=0, B=1, C=2, D=3 дает таблицы, из которых видно, что для элемента 2 по операции умножения отсутствует обратный (в таблице умножения отличных от нуля элементов появился нулевой элемент A). Таким образом, кольцо Z_n является полем только тогда, когда n – простое число. Таблицу умножения поля GF(4) можно исправить. Над полем GF(2) есть только один неприводимый многочлен X^2+X+1, соответственно, $GF(4) = GF(2)[X]/X^2+X+1$ (см. раздел про поля Галуа).

Используемое для колец обозначение берет начало в практике образования расширенных полей. Например, поле комплексных чисел обозначают так $C = R(X)/(X^2–1)$. Это означает то, что для образования такого поля используется внешний по отношению к полю вещественных чисел R(X) элемент i, корень неразрешимого уравнения $X^2–1=0$.

Уравнение неразрешимо, ну, что же. Его корень мы вовлекаем в игру, получаем расширенное поле с элементами $a+ib$ (или кольцо).

Напомним, что поле Галуа A=GF(p) с элементами вида $C=[c_0, c_1, ..., c_{n–1}]$, коэффициентами полиномов, конструируют при помощи дополнительного полинома, позволяющего "замкнуть операции на себя". После чего можно пользоваться привычными нотациями арифметических операций над элементами – полиномами. Операции нижнего уровня над коэффициентами, например, бинарными, соотносят с более просто устроенным обслуживающим поле верхнего уровня конечным полем. В работе участвуют одновременно два поля.

Циклические подгруппы и кольца. В связи с конечным полем A=GF(p) стоит называть и различать между собой субконструкции мультипликативных циклических групп GF(p)* и циклических колец $R = A[x]/(x^n–1)$.

В группе GF(p)* последовательные степени некоторого элемента, т.е. продукты умножения некоторого *примитивного* элемента g на самого себя, образуют, как известно, циклическую подгруппу $g^0, g^1, g^2, ..., g^{v-1}$, где v – количество элементов подгруппы.

В кольце R = A[x]/(x^n–1) последовательные произведения некоторого элемента g (*генератора*) на элемент кольца x, образуют циклическое подкольцо $gx^0, gx^1, gx^2, ..., gx^{v-1}$ (что соответствует циклическому сдвигу коэффициентов вправо), где v – количество элементов подгруппы.

В качестве генератора рассматривается обычно некоторый многочлен минимальной степени подкольца – он же делитель многочлена x^n–1.

Циклические коды, элементы циклического кольца, в общем, не ортогональны. Это в особенности касается кодов, оперирующих на нижнем уровне переменными {1, –1}, так как есть гипотеза Ризера об отсутствии циклических матриц Адамара степени, большей 4. В теории кодирования для классификации кодов вводится понятие *кодового расстояния* Хэмминга. В зависимости от расстояний между элементами, выделяют коды информирующие об ошибках или коды исправляющие одну, две и более ошибок.

Идеалы. Подгруппы или подкольца – множества, замкнутые относительно операции умножения на элемент группы или кольца. Тем самым, группа или кольцо распадается на образования, более крупные, чем элементы, их называют идеалами. Идеал, который не совпадает со всем кольцом, называется *собственным*, а совпадающий – главным идеалом. Специфическими единичными метками идеалов, их идентификаторами (как для ремня, пряжка), являются порождающие их элементы (генераторы).

Из определения идеала следует, что во всяком кольце идеалами являются само кольцо и нуль-идеал (нулевой элемент). Кольцо, не содержащее других идеалов, кроме этих двух, называется простым. Все тела и поля являются простыми кольцами.

Если кольцо не содержит элемент поля, то этот элемент содержится в другом кольце. Все они вложены в *главное кольцо* R = A[x]/(p), $p(x)$ – неприводимый полином. Элементы коммутативного кольца – линейные комбинации степеней x с коэффициентами $c_0, c_1, ..., c_{n-1}$, определенными, как ранее отмечалось, в обслуживающем операции верхнего уровня поле.

Неприводимость – невозможность выразить корни полинома элементами обслуживающего операции поля. Неприводимость не означает, что не выражаемый непосредственно корень никак не связан связан с элементами обслуживающего поля и для него нельзя написать пусть формальной, но формулы. Пример у нас перед глазами, когда мы вкладываем некоторое содержание в формулу i = sqrt(–1).

С четверкой 2=sqrt(4) функция возвращает вполне себе не выходящее за пределы поля значение. Квадрат мнимой единицы равен вещественному числу. Корень квадратный – это функция (гомоморфизм), содержание которой зависит от элементов, к которой она прилагается.

В теории чисел аналогами идентификаторов идеалов являются *взаимно простые числа*. Впервые понятие идеалов (вкладывая несколько иное содержание) предложил Кумер, работавший над обобщением целых чисел Гауссом – комплексных чисел с целочисленными составляющими. Позднее абстрактные алгебраические структуры: кольца, идеалы и модули популяризировал Дедекинд, принявший кафедру Гаусса. Исследования Дедекинда были изданы в виде приложения к «Теории чисел» Дирихле.

Основная идея, ради чего были введены идеалы, – если не сами алгебраические числа, определенные в расширенных полях, то хотя бы идеалы, более аморфные числовые конструкции, могут быть однозначно выражены через "произведение" простых идеалов. Что позволяет ввести некоторое осмысленное абстрактное деление.

7. ПОЛЯ ГАЛУА GF(2^m)

Поле GF(2^m) имеет 2^m элементов, нумеруемых как 0, 1, 2,..., 2^m–1. При бинарном представлении под каждое целое число требуется m бит. Биты можно использовать как коэффициенты полиномов порядка не более m–1.

Представление элементов GF(2^3). Элементы поля Галуа GF(2^3)=GF(8) можно задавать либо их индексами 0, 1, ..., 8, бинарным 000, 001, 010, ..., 111 или полиномиальным 0, 1, A, 1+A, ..., A^2+A+1 представлениями.

При помощи индексов несложно задать таблицы сложения S и умножения M, отвечающих примитивному ниже полиному A^3+A+1 (с помощью которого производится "возвращение" старших степеней). Сложение и умножение производится обращением к клеткам таблицы, не затрагивая полиномиальной сущности данных.

```
gfinit(8);
A=3; B=5; puts("A="+A+" B="+B);
C=gfadd(A,B); puts("C=A+B="+C);
B=gfsub(C,A); puts("B=C-A="+B);
D=gfmul(A,B); puts("D=A*B="+D);
B=gfdiv(D,A); puts("B=D/A="+B);
```

Результат: A=3, B=5, C=A+B=6, B=C-A=5, D=A*B=4, B=D/A=5.

Вычислим значения примитивного полинома A^3+A+1 в поле Галуа GF(8), его корни (примитивные элементы) видны в порожденном им поле. В общем случае, корни произвольного полинома размещены за пределами поля.

```
gfinit(8);
for (A=0;A<8;A++) {
A3=gfmul(gfmul(A,A),A);
f=gfadd(gfadd(A3,A),1);
puts("A="+A+" A^3+A+1="+f);
}
```

Результат: A=0, A^3+A+1=1; A=1, A^3+A+1=1; A=2 A^3+A+1=0 и т.п.

Пояснение. В *полиномиальной арифметике* сложение и вычитание, по модулю p, в том числе, заведомо не повышает степени полинома. Проблемно лишь умножение. Его результат, чтобы вернуть показатель старшей степени обратно, делят на некоторый *неприводимый многочлен* степени m и находят остаток от деления. Единственная сложность: когда коэффициенты полиномов заданы в поле GF(p), расчет остатка проводится с учетом арифметики алгебраического сложения по модулю, расчет дает полиномы с положительными коэффициентами.

При работе с бинарными коэффициентами, например, характеристика поля p=2. Сложение полиномов сводится к побитовому сложению их коэффициентов по модулю 2 (XOR), регламентирующему 1+1=0, т.е. при сложении x^2+x с собой получим 0. При m=8 коэффициенты полинома займут байт информации. Важно, чтобы и операция умножения не выводила число коэффициентов итогового полинома за пределы этого байта.

Пример 4. Обратимся к случаю GF(2^3), p=2, m=3. Найдем "возвратное значение" произведения $x^2?(x^2+1)=x^4+x^2$, для чего разделим его на неприводимый полином x^3+1: вычитание из x^4+x^2 приближающего его произведения $(x^3+1)x=x^4+x$ в поле GF(2) даст остаток x^2+x (обходимся без минуса). Следовательно, $x^2?(x^2+1)=x^2+x$.

Таблица умножения может быть составлена заранее. Пометим маркером в таблице 2 коэффициенты полиномов, которые мы перемножали, и соответствующий результат.

Таблица 2. Таблица умножения

	000	001	010	011	100	101	110	111
000	000	000	000	000	000	000	000	000
001	000	001	010	011	100	101	110	111
010	000	010	010	110	001	011	101	111
011	000	011	110	101	101	110	011	000
100	000	100	001	101	010	110	011	111
101	000	101	011	110	110	011	101	000
110	000	110	101	011	011	101	110	000
111	000	111	111	000	111	000	000	111

Если в качестве *порождающего* использован многочлен f(x), на который делится многочлен x^m+1, то в этом поле умножение на x соответствует циклическому сдвигу коэффициентов. Циклические коды, возникающие здесь, используются в теории защитного кодирования.

8. ИЗ ИСТОРИИ

Задачи трисекции угла, удвоения куба, квадратуры круга и т.п. – древние "безнадежные" вопросы, в процессе рассмотрения которых накоплено немало полезных наблюдений. Теория конечных полей Галуа также оказалась многоплановым приобретением.

Предшественником Галуа был Лежандр, чья книга "Теория чисел" вызвала в нем живой резонанс и интерес к теме разрешимости полиномиальных уравнений. Среди полиномиальных уравнений есть простейшие, связанные с определением корней квадратных и прочих корней (радикалов). Выход радикалов за пределы области определения коэффициентов полинома был обнаружен давно, история корня квадратного из двух насчитывает тысячелетия. Этот корень не передать отношением целых (или рациональных, все равно) чисел, т.е. радикалами более низкого порядка, корнями линейных уравнений.

Гаусс, старший современник Галуа, выразил корни уравнения $x^{17}–1=0$ всего лишь через корни квадратные, допуская возможность вложения (корень из корня). Отсюда следует возможность вписать в круг правильный многоугольник с 17-ю сторонами с помощью циркуля и линейки, поскольку уравнения второго порядка разрешимы графически. В сети Интернет сегодня есть анимированные построения этого знаменитого многоугольника. Галуа отличился тем, что ввел (почти очевидно востребованное для определения не коэффициентов, а искомых решений) новое понятие конечного поля: поля коэффициентов полинома, расширенного корнями. Например, точки плоскости, до которых мы можем добраться с помощью циркуля и линейки, достаточны для построения конечного поля размерности степеней двойки. Поэтому в круг можно вписать многоугольник с 4+1=5 и 16+1=17 сторонами. А семиугольник нельзя.

Теорема Гаусса может быть интерпретирована как положение о разрешимости числа $\cos(2\pi/n)$ в арифметике, в которой фигурируют корни квадратные, не выше, только если $n=2^k\times$(произведение чисел Ферма). Иначе эти числа неразрешимы.

Допустим, хлопоты с дефиницией радикалов – корней простейших полиномиальных уравнений – завершены их аксиоматическим определением. Кончаются ли на этом проблемы с конечностью манипуляций для достижения решений полиномиальных уравнений? Да, если поля имеют простую структуру, при которой задача поиска решения сводима к этим самым радикалам. В отличие от случая Гаусса, речь идет о радикалах более высокого порядка, с которыми мы "справились". Корни уравнения $8x^3–6x+1=0$ неразрешимы через корни квадратные, но на помощь приходит корень кубический. Для $x^5–4x+2=0$ не поможет и корень пятой степени. Иерархия решений сложных уравнений высоких порядков иная, чем у уравнений до порядка 4. До Галуа это доказали Абель и Руфини.

Иными словами, для порядков 2, 3, 4 есть список приемов, при помощи которых самое общее уравнение той же степени 2, 3, 4 может быть сведено к решению нескольких вспомогательных уравнений простейшей формы. Для уравнения пятой степени этого сделать нельзя, можно доказывать, как выше, на конкретных примерах.

Исторически в этой теории возникли перестановки корней, когда осознали, что замкнутая на комплексной плоскости траектория изменения любого параметра уравнения приводит к изменению корней, имеющему параметрические точки, когда корни "ротируются" местами. Сходно ведет себя любая функция от этих корней. Радикалы, корни n-й степени, расположены симметрично на комплексной плоскости, группа их перестановок циклическая, разрешимая. Для уравнения степени пять разрешимость группы перестановок корней, в общем, приводит к противоречию, в связи с чем радикалов для их выражения недостаточно. Для разрешимости нужно вводить дополнительные уравнения, что "не страшно, но хлопотно".

У циклических матриц Адамара, корней матричного квадратного уравнения $\mathbf{H}^{\mathrm{T}}\mathbf{H}=n\mathbf{I}$ при ограничении элементов 1 и –1, есть сходное пороговое значение порядка: задача разрешима для матриц не выше четвертого порядка. Положение, причем, признано недоказанным.

В нынешней утвердившейся систематизации знаний теория Галуа оперирует подстановками, элементами симметрических групп S_n, сводя задачу разрешимости полиномиальных уравнений к задаче построения цепочки вложенных друг в друга так называемых нормальных подгрупп, завершающейся тривиальной подгруппой. Для уравнения 5-го порядка группа симметрий поля, порожденного корнями многочлена, изоморфна группе вращении додекаэдра (или икосаэдра, фигуры двойственны); но у этой группы проблема с нормальными подгруппами. Если тривиальная подгруппа не достижима, значит иерархия решений уравнений сложна, среди решений есть такие, которые не выражаются через радикалы.

Это важно? Да, в общем, нет. Алгебраических конструкций очень много. Одни только определения понятий занимают в обзорах солидное место, не говоря о примерах оперирования с ними. Ранее нас интересовала уже не столько центральная задача, решаемая Галуа, сколько прикладные примеры на построение ортогональных матриц, с которыми инструментарий, наработанный в процессе изучения групп Галуа, отлично справляется.

В качестве литературы настоящее учебное пособие включает материал книг последних лет, размещенных в библиотеке ГУАП [1–3], а также два учебных сайта университета [5, 6]. На них есть доступ к обширному перечню статей авторов на тему нахождения матриц ортогональных преобразований [6–12], дополняющих изложенный в пособии материал.

Литература

1. Введение в цифровую обработку изображений: Методы фильтрации и сжатия изображений: учебное пособие / М. Р. Гильмутдинов [и др.]; С.-Петерб. гос. ун-т аэрокосм. приборостроения. – СПб.: Изд-во ГУАП, 2015. – 76 с.
2. Умняшкин В.. М. Теоретические основы цифровой обработки и представления сигналов. Учебное пособие – С: ФОРУМ: ИНФРА-М, 2014. – 304 с.
3. Цифровая обработка изображений Р. Гонсалес, Р. Вудс; пер. Л. И. Рубанов, пер., ред. П. А. Чочиа. - 3-е изд., испр. и доп. – М.: Техносфера, 2012. – 1104 с.
4. Учебный сайт по дисциплине ортогональные преобразования, livelab.spb.ru (время обновления 18.01.1918)
5. Учебный сайт по дисциплине матрицы Адамара, mathscinet.ru (время обновления 18.01.1918)
6. Балонин Н. А., Сергеев М. Б., Суздаль В.С. Динамические генераторы квазиортогональных матриц семейства Адамара // Труды СПИИРАН. 2017. Вып. 5(54). С. 224–243.
7. Балонин Н. А., Сергеев М. Б. Матрицы локального максимума детерминанта // Информационно-управляющие системы. 2014. № 1. С. 2–15.
8. Балонин Н. А., Сергеев М. Б. Расширение гипотезы Райзера на двуциклические структуры и разрешимость матриц Адамара орнаментом в виде бицикла с двойной каймой // Информационно-управляющие системы. 2017. № 1. С. 2–10.
9. Балонин Н. А., Сергеев М. Б. Матрицы Мерсенна и Адамара // Информационно-управляющие системы. 2016. № 1. С. 2–15.
10. Балонин Н. А., Сергеев М. Б. Матрицы Мерсенна и Адамара, произведения // Информационно-управляющие системы. 2016. № 5. С. 2–14.
11. Балонин Н. А., Сергеев М. Б. К вопросу существования матриц Адамара и Мерсенна // Информационно-управляющие системы. 2013. № 5. С. 2–8.
12. Балонин Н. А., Сергеев М. Б. Нормы обобщенных матриц Адамара // Вестник СПбГУ. Сер. 10. 2014. Вып. 2. С. 5–11. (Вестник СПбГУ)